

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт

ОБ'ЄКТНО – ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

для студентів 2 курсу
спеціальності– комп'ютерні науки

Затверджено на методичній комісії

Факультету комп'ютерних наук

Протокол № ____ від _____

Декан _____ Коваленко Л.Б.

Затверджено на засіданні кафедри

Інформаційних технологій

Протокол № ____ від _____

Зав. Кафедрою _____ Кузніченко С.Д.

ОДЕСА 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт

ОБ'ЄКТНО – ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

для студентів 2 курсу
спеціальності – комп'ютерні науки

Затверджено на методичній комісії
Факультету комп'ютерних наук
Протокол № _____ від _____

ОДЕСА 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт

ОБ'ЄКТНО – ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

для студентів 2 курсу

напрямку підготовки – комп'ютерні науки

МЕТОДИЧНІ ВКАЗІВКИ до виконання лабораторних робіт з навчальної дисципліни ОБ'ЄКТНО – ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ для студентів 2 курсу денної форми навчання спеціальності – комп'ютерні науки.

Укладачі:

Рольщіков В. Б., старший викладач кафедри інформаційних технологій,
Штефан Н. З., асистент кафедри інформаційних технологій
ОДЕкУ, 2018, 104 с., укр. мова.

ЗМІСТ

Вступ.....	Error! Bookmark not defined.
1. Мета та задачі лабораторних робіт	8
2. Індивідуальні завдання.....	9
3. Методичні рекомендації	9
4. Форма і методи контролю.....	10
ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ ТА ОХОРОНИ ПРАЦІ	12
ЛІТЕРАТУРА	12
Лабораторна робота №1	14
Рекомендації до виконання роботи:	14
Постановка завдання та вихідні дані для варіантів:.....	40
ЛАБОРАТОРНА РОБОТА №2	46
Рекомендації до виконання роботи (частина 1):.....	46
Рекомендації до виконання роботи (частина 2): моделювання за допомогою плагіну Parurys.	52
Постановка завдання і вихідні дані для варіантів:	60
Варіант 1.....	61
Варіант 2.....	62
Варіант 3.....	62
Варіант 4.....	64
Варіант 5.....	65
Варіант 6.....	66
Варіант 7.....	66
Варіант 8.....	68
Варіант 9.....	69
Варіант 10.....	69
ЛАБОРАТОРНА РОБОТА №3	71
«Створення UML – діаграми для Спадкування членів класу. Перевизначення методів».....	71
Рекомендації до виконання роботи (частина 1):.....	71

Рекомендації до виконання роботи (частина 2): моделювання за допомогою плагіну Rarurys.....	75
Постановка завдання і вихідні дані для варіантів:	79
Варіант 1.....	80
Варіант 2.....	81
Варіант 3.....	82
Варіант 4.....	83
Варіант 5.....	83
Варіант 6.....	84
Варіант 7.....	85
Варіант 8.....	86
Варіант 9.....	86
Варіант 10.....	87
Контрольні питання	88
ЛАБОРАТОРНА РОБОТА №4	88
« Створення UML-діаграми для відображення абстрактних класів та їх реалізації».....	88
Рекомендації до виконання роботи:	88
Рекомендації до виконання роботи (частина 2): моделювання за допомогою плагіну Rarurys.....	92
Постановка завдання і вихідні дані для варіантів:	98
Варіант 1.....	99
Варіант 2.....	100
Варіант 3.....	100
Варіант 4.....	101
Варіант 5.....	102
Варіант 6.....	102
Варіант 7.....	103
Варіант 8.....	103
Варіант 9.....	104

Варіант 10.....	104
Контрольні питання	104

ВСТУП

Дисципліна «Об'єктно-орієнтоване програмування» є загальноосвітньою. Метою лабораторних робіт з цієї дисципліни є закріплення теоретичного лекційного матеріалу та надбання практичних навиків у розробці та налагодженні програмного забезпечення студентами, а також формування у студентів практичних навичок об'єктно-орієнтованого програмування різноманітних задач в середовищі програмування JAVA 2 sdk v 1.4 з використання інтегрованого середовища розробки Eclipse.

Лабораторні роботи з дисципліни «Об'єктно-орієнтоване програмування» проводяться на базі учбових лабораторій кафедри інформаційних технологій із застосуванням відповідного програмного та апаратного забезпечення.

Лабораторні роботи призначені для студентів **II-го курсу денної форми навчання**, що навчаються за спеціальністю 122 «Комп'ютерні науки».

1. Мета та задачі лабораторних робіт

Метою лабораторних робіт є закріплення знань, здобутих на лекційних заняттях курсу з відповідної дисципліни, та набуття практичних навичок з роботи у сучасних середовищах розробки програм.

У результаті виконання програми практики студент повинен:

- **знати**: принципи будовання об'єктів, їх властивості (інкапсуляцію, спадковість та поліморфізм), елементи програмування мовою об'єктного програмування: JAVA, структуру інтегрованого середовища розробки програм Eclipse;

- **уміти**: складати програми із застосуванням засобів об'єктного програмування, що їх має мова JAVA, розробляти власні, спадкувати та видозмінювати, вже існуючі об'єкти, звертатися до приватних полів та методів об'єктів, оформляти віконний інтерфейс програми, підключати до нього

обробники подій, обробляти виключні ситуації, що можуть виникнути у програмі.

2. Індивідуальні завдання

Індивідуальні завдання кожного студента обумовлюються тим варіантом завдань, котрий він одержав від викладача. Викладач зобов'язаний слідкувати за тим, щоб студент не міг змінити цей варіант та виконувати інше завдання.

3. Методичні рекомендації

Для виконання лабораторних робіт за студентом на постійно закріплюється комп'ютер та всі завдання конкретизуються відповідно до нього. Викладач повинен ознайомити студентів з положеннями даних методичних вказівок та роздати відповідну навчальну та довідкову літературу, якщо це потрібно, навіть у електронному варіанті.

Завдання лабораторних робіт повинні виконуватись студентом тільки у межах навчального класу на тому комп'ютері, що закріплений за ним. Додаткова робота може проводитись також і самостійно студентом поза навчальними класами, користуючись допомогою відповідної довідкової літератури та інтерактивних довідкових систем. При цьому варто користуватися тими знаннями, що здобуті при вивченні навчальних курсів, “Основи програмування та алгоритмічні мови”, “Об’єктно-орієнтоване моделювання” та “Об’єктно-орієнтоване програмування”.

Під час знаходження у приміщенні, де виконуються лабораторні роботи, студент повинен строго дотримуватись правил техніки безпеки та санітарії при роботі за комп'ютером, а також, правил пожежної безпеки у відповідних приміщеннях.

4. Форма і методи контролю

До початку лабораторного заняття студент, за рахунок годин, відведених на самостійну роботу, повинен підготуватися до виконання роботи. Підготовка включає: ознайомлення з метою роботи, вивчення теоретичної частини, що відноситься до неї, оформлення титульного аркуша та 1-го і 2-го пунктів протоколу.

Вимоги до протоколу

Протокол про виконання лабораторної роботи повинен містити:

1. умову задачі;
2. UML діаграму, яка відобразатиме основні конструкції, що використовуються при вирішенні даної задачі;
3. лістинг програми, де в якості коментарів повинні виступати П.І.Б. студента, № групи і варіант завдання, а також додаткові коментарі, що описують призначення методів, конструкторів і т.д.;
4. результати роботи програми;
5. висновки по даній роботі;

Перед проведенням заняття викладач зобов'язаний перевірити в студента наявність підготовленого і заповненого протоколу. Якщо протокол відсутній, студент не допускається до виконання роботи з виставлянням нульової оцінки за практичну частину роботи. Після перевірки наявності й правильності заповнення протоколу викладач проводить опитування знань студента й допускає його до роботи з відповідною оцінкою за теоретичну частину.

Опитування знань з теоретичної частини проводиться за допомогою контрольних питань до даної лабораторної роботи, повна, якісна відповідь на них гарантує студентові отримання 100 балів за теоретичну частину роботи.

Практична частина роботи оцінюється в такий спосіб. На кожному занятті викладач оцінює відсоток виконаної роботи. Якщо лабораторна робота розрахована на чотири і більше годин, то загальна оцінка роботи прямо пропорційна відсотку виконання на першому занятті.

Наприклад, якщо на першому занятті студент написав 60% програмного коду, а на іншому 40%, що залишилися, то здавши на третьому занятті повністю оформлений протокол, він отримує 100% балів за дану роботу. Навпаки, якщо на першому занятті виконає 10% завдання, на другому 20%, а на завершальному занятті зробив протокол з повністю виконаною лабораторною роботою, то студент може розраховувати на отримання тільки 30% від можливих балів за практичну частину роботи. На оцінку практичної частини впливає й термін здачі протоколу.

Для отримання максимуму балів повністю оформлений протокол повинний бути зданий на останньому занятті, відведеному для виконання роботи або наступного дня після виконання роботи. Затримка зі здачею протоколу знижує оцінку на 10% за кожен тиждень затримки. Тобто, студент, який за ритмічність виконання роботи міг отримати, наприклад, 70% можливих балів, але без поважної причини здав протокол через 16 днів після її завершення, отримає тільки $70 - 30 = 40\%$ балів за виконання роботи. Максимальна кількість балів за практичну частину роботи становить 100%.

Підсумкова оцінка за виконання кожної лабораторної роботи розраховується знаходженням середнього арифметичного між практичною та теоретичною частиною роботи. Тобто, максимальна оцінка за кожну роботу 100 балів. Допуск до заліку або іспиту можливий тільки при отриманні не менше 50% балів за виконання всіх запланованих лабораторних робіт. Виконання першої половини робіт на 100%, а другої на 40% оцінюється тільки в 40% і, таким чином, студент не отримує допуск до підсумкового контролю знань.

Після виконання роботи студенти захищають свої звіти викладачеві. Виконання лабораторної роботи оцінюється за двобальною системою «зараховано», «не зараховано» з градаціями у відсотках відповідно до «Положення про проведення підсумкового контролю знань студентів».

ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ ТА ОХОРОНИ ПРАЦІ

Кожен студент перед початком навчальної практики мусить вивчити правила роботи з комп'ютерною технікою в лабораторіях кафедри інформаційних технологій, пройти співбесіду з інструктором з техніки безпеки та розписатися в журналі з техніки безпеки.

ЛІТЕРАТУРА

Базова

1. Императивное программирование объектно-ориентированное моделирование: Java, UML, OCL: учебное пособие для студентов высших учебных заведений / А.Ф.Верлань, И.А.Чмырь, С.Д.Кузниченко, Л.Б.Коваленко. – Одесса: Экология, 2013. – 432 с.

2. Шилдт Г. Java 8. Полное руководство; 9-е изд.: Пер. с англ. – М. : ООО "И.Д. Вильямс", 2015. – 1376 с.: ил.

Допоміжна

1. Specification: JSR-337 Java® SE 8 Release Contents ("Specification"), Version: 8, Final Release: March 2014 – Redwood City: Oracle America Inc., 2014. – 760 p.

2. Эккель Б. Философия Java. Библиотека программиста. 4-е изд. СПб.: Питер, 2009. – 640 с.: ил.

3. Вайсфельд М. Объектно-ориентированное мышление. – СПб.: Питер, 2014. – 304 с.: ил.

4. Java. Методы программирования: уч.-мет. пособие / И.Н.Блинов, В.С.Романчик. – Минск: издательство «Четыре четверти», 2013. – 896 с.

Лабораторна робота №1

«Класи і об'єкти. Способи ініціалізації полів».

Мета: Навчитися описувати класи, що містять статичні і нестатичні компоненти, задавати значення полів прямим зверненням до них і за допомогою блоків ініціалізації, створювати об'єкти, масиви об'єктів і працювати з ними.

Рекомендації до виконання роботи:

Класи та їх представники

Клас є основною, функціонально закінченою структурною одиницею, яка задає шаблон, за яким будуть будуватися реальні об'єкти, що є представниками цього класу. У простому випадку клас складається з:

- набору полів, які можна розглядати як змінні або константи різних типів даних з числа тих, що входять до складу мови Java або мають тип раніше створених класів;
- набору методів, які можна розглядати як функції, які повертають "return" деяке значення зазначеного типу даних, або як процедури, які нічого не повертають "void", але виконують перелік операторів, що в них входять.

Поля і методи можуть бути статичними і нестатичними. Різниця полягає в тому, що статичні компоненти є приналежністю класу, існують в єдиному числі і звернення до них проводиться шляхом вказівки імені цього класу і через оператор "крапка" імені компонента (поля або методу), а набір нестатичних компонентів є приналежністю об'єкта, причому кожен об'єкт при створенні його оператором "new" отримує свою персональну копію всього нестатичного складу класу, доступ до якого здійснюється шляхом зазначення імені об'єкта і через "крапку" імені компонента.

```

class pencil { int length;
    String color = "";
    static String produser; void lookInfo () {
        System.out.println ("This is a" + color + "pen" + length +
"cm length.")
    }
    int getLength () {
        return length;
    }
    int summaryLength (pencil pl) {
        return length + pl.length;
    }
    static void showInfo (pencil pl) {
        System.out.println ("This is a" + color + "pen" + length
+ "cm length.");
    }
    static String getColor (pencil pl) {
        return pl.color;
    }

    public static void main (String args []) {
        pencil My_pen = new pencil ();
        pencil Your_pen = new pencil ();
        pencil.produser = "Kooh-I-Noor";
        My_pen.lookInfo ();
        pencil.showInfo (Your_pen);
        System.out.println ("Summary length =" + My_pen.summaryLength
(Your_pen)); System.out.println ("My pencil length =" +
My_pen.getLength ());
        System.out.println ("Your pencil color is" + pencil.getColor
(Your_pen)); System.out.println ("Their produser" +
pencil.produser);
    }
}

```

У наведеному прикладі показаний склад класу "олівець", за допомогою якого можна примітивно промоделювати деякі практичні додатки:

- `length` і `color` - нестатичні поля класу `pencil`, причому `length` позамовчуванню буде ініціалізований нулем, а `color` - величиною `null`, а оскільки при спробі вивести її вміст на екран командою `System.out.println (color)` буде згенерована помилка виконання, то необхідно безпосередньо проініціалізувати її значенням порожнього рядка. У об'єктів класу `pencil` `My_pen` і `Your_pen`, які створені в методі `main` є свій набір цих змінних, звернутися до яких можна `My_pen.length` або `Your_pen.color`;
- `producer` - статична змінна "виробник", в методі `main` їй присвоюється значення `pencil.Producer = "Kooh-I-Noor"` і проглядається її вміст `System.out.println ("Their producer" + pencil.Producer)`;
- `lookInfo` - метод (процедура) для виводу на екран інформації про поля об'єкта, для якого вона викликана. Оскільки вона нестатична, то і викликається вона не для класу, а для конкретного представника, наприклад у методі `main`:

```
My_pen.lookInfo ();
```

- `getLength` - метод (функція), що повертає величину типу `int` - довжина олівця. Вона також нестатична і, тому також як і `lookInfo ()` є на складі створених представників і викликається для них, наприклад в методі `main`:

```
System.out.println ("My pencil length =" + My_pen.getLength ())
```

Зверніть увагу, що функція (на відміну від процедури) не може бути використана сама по собі, вона повинна брати участь в деякому виразі; • `summaryLength` - також як і `getLength` є нестатичною функцією (обчислює суму довжин двох олівців). У тілі функції беруть участь два поля: одне `length` - для поточного об'єкта і друге `pl.length` - для переданого їй аргументу типу `pencil`;

- `showInfo` - статична процедура, яка є аналогом `lookInfo`, але існує тільки у складі самого класу, а не його представників, тому і викликається для нього в методі `main`:


```
pencil.showInfo (Your_pen);
```

 причому виводить на екран вона інформацію для того об'єкта, який переданий їй як аргумент;
- `getColor` - статична функція, що повертає величину типу `String` - значення поля `color` для об'єкта, зазначеного в аргументі. У методі `main` вона бере участь у виразі:

```
System.out.println ("Your pencil color is" + pencil.getColor (Your_pen));
```

Метод `main` - завжди статичний, він входить до складу класу в тих випадках, коли передбачається можливість запуску даного класу на виконання виконуючим середовищем Java. В даному випадку в ньому створені об'єкти класу:

```
pencil pencil My_pen = new pencil ();
pencil Your_pen = new pencil ();
```

і з ними виконані всі перераховані вище дії.

Використання блоків ініціалізації

Блок ініціалізації - особлива конструкція в складі класу, призначена для ініціалізації полів. Вони можуть бути як статичними, в цьому випадку в них можуть задаватися значення статичних полів класу, так і нестатичними, в цьому випадку вони входять до складу кожного створеного об'єкта класу і можуть ініціалізувати поля даного об'єкта.

У тілі блока можуть виконуватися операції присвоювання, викликатися різні методи, а також використовуватися різні оператори, у тому числі розгалуження і цикли.

Наприклад, у випадку з олівцями, можна видозмінити клас pencil, додавши до його складу блок ініціалізації, в якому кожному парному олівцю, що створюється, буде присвоєно колір "червоний", а кожному непарному - "синій". Крім того довжину олівців будемо задавати 10 см з урахуванням деякої похибки (+ / -) 1 см. Для контролю кількості створюваних об'єктів додамо лічильник - статичну змінну counter.

```
class pencil {
    int length;
    String color;
    static String producer;
    static int counter;
    {
        length = 10 + (int) (Math.random () * 2-1);
        if (counter% 2 == 0) color = "Red";
        else
            color = "Blue"; counter + +;
    }
}

// Все інше без зміни
public static void main (String args []) {
    pencil.counter = 10;
    pencil pen1 = new pencil ();
    pen1.lookInfo ();
    pencil pen2 = new pencil ();
    pen2.lookInfo ();
    pencil pen3 = new pencil ();
    pen3.lookInfo ();
    System.out.println ("Next counter value is" + pencil.counter);
}
```

В даному випадку лічильник був завантажений стартовою величиною 10, щодо якої і пішов відлік. Оскільки блок ініціалізації нестатичний, то він автоматично викликається для кожного створюваного об'єкту `new pencil ()`, при цьому полям `length` і `color` цього об'єкту присвоюються значення відповідно до задуманого принципу, а статичний лічильник класу `counter` інкрементується, що дозволяє в конкретний момент часу знати кількість раніше створених об'єктів.

Проконтролювати роботу блоку ініціалізації можна викликом процедури `lookInfo` для кожного зі створених представників, а значення лічильника - зверненням до поля класу `pencil.counter`.

Робота з масивами об'єктів

При створенні великої кількості об'єктів, для яких в майбутньому передбачається робота в циклі (перебір, перегляд, сортування і подібне) зручно організувати їх у вигляді масиву з елементами типу даного класу. В даному випадку це буде виглядати так:

```
pencil Pen [] = new pencil [25].
```

Даний запис говорить про те, що оголошений масив з 25-ти елементів типу `pencil`. Для того, щоб можна було працювати з елементами масиву, кожен з них необхідно проініціалізувати

```
pen [i] = new pencil (),
```

що зручно організувати в циклі. Таким чином, роботу блоку ініціалізації створеного вище можна було б перевірити на більшій кількості елементів досить елегантно:

```
public static void main (String args []) {
    pencil.counter = 10;
    pen [] = new pencil [25];
    for (int i = 0; i < pen.length; i + +)
```

```

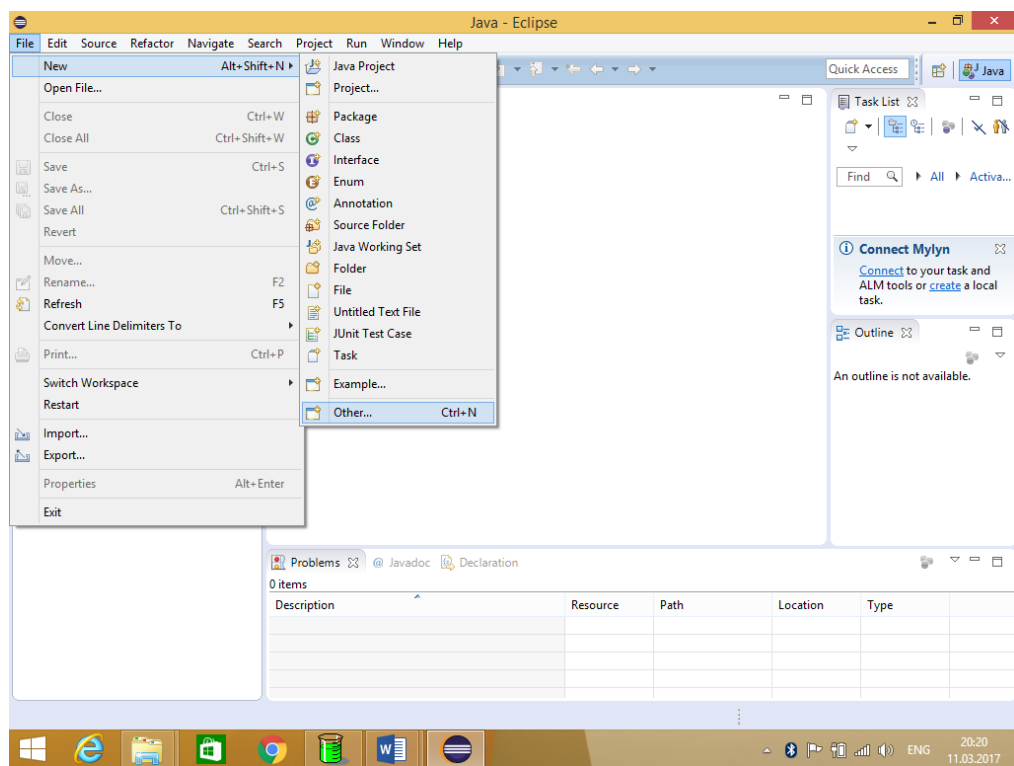
    {
        pen [i] = new pencil (); pen [i]. lookInfo ();
    }
}

```

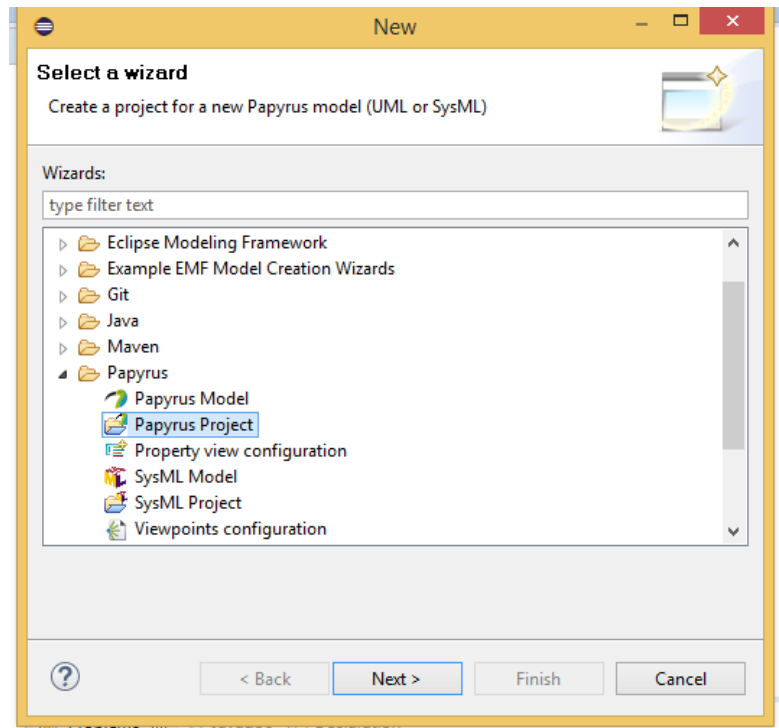
Рекомендації до виконання роботи (частина 2): моделювання за допомогою плагіну Rapurys.

На початку роботи необхідно створити проект Rapurys Project у Eclipse.

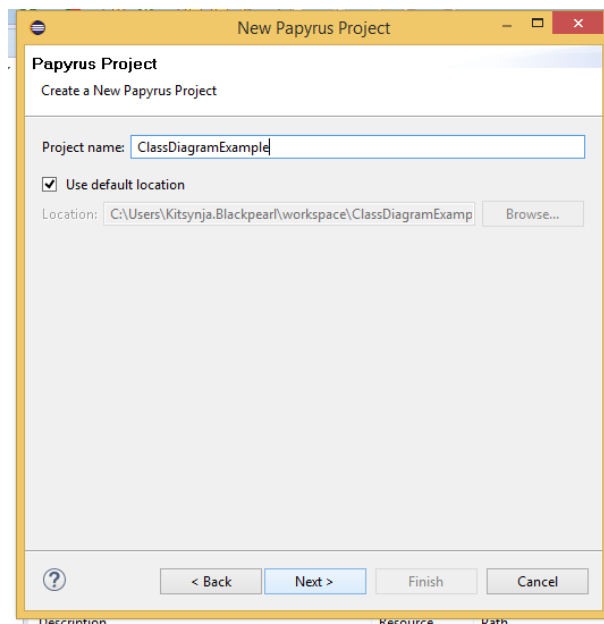
Для цього заходимо у меню File->New->Other:



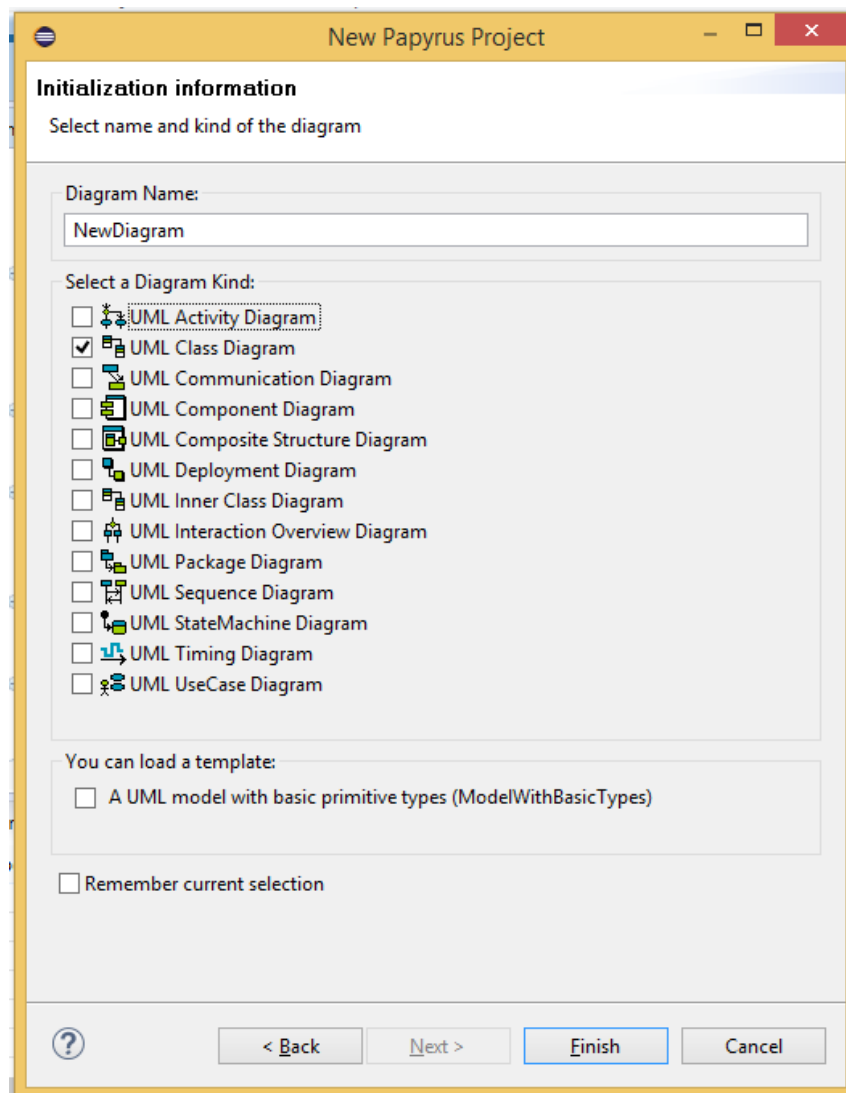
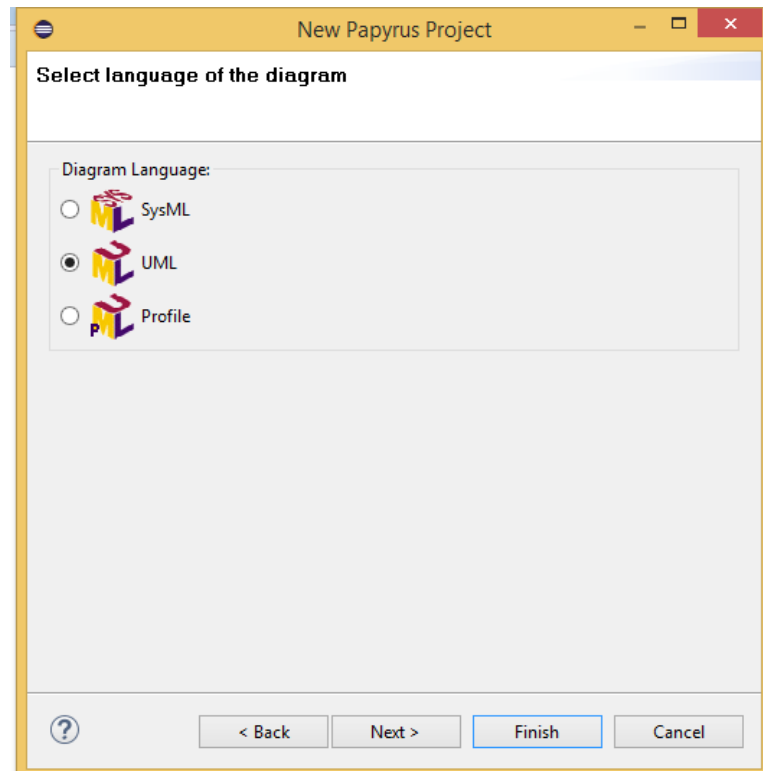
У відкритому вікні New потрібно обрати Rapurys Project:



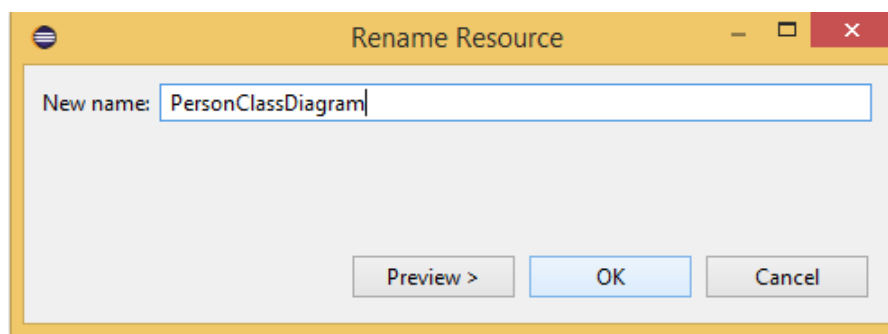
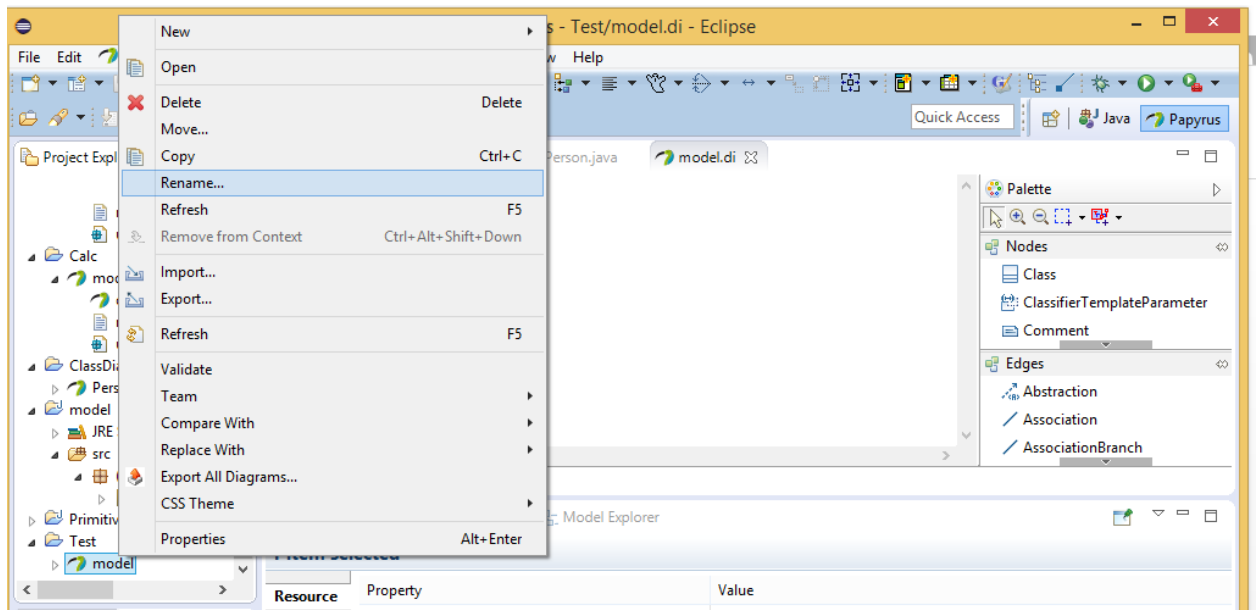
та вказуємо ім'я майбутньої діаграми класів наприклад `ClassDiagramExample`



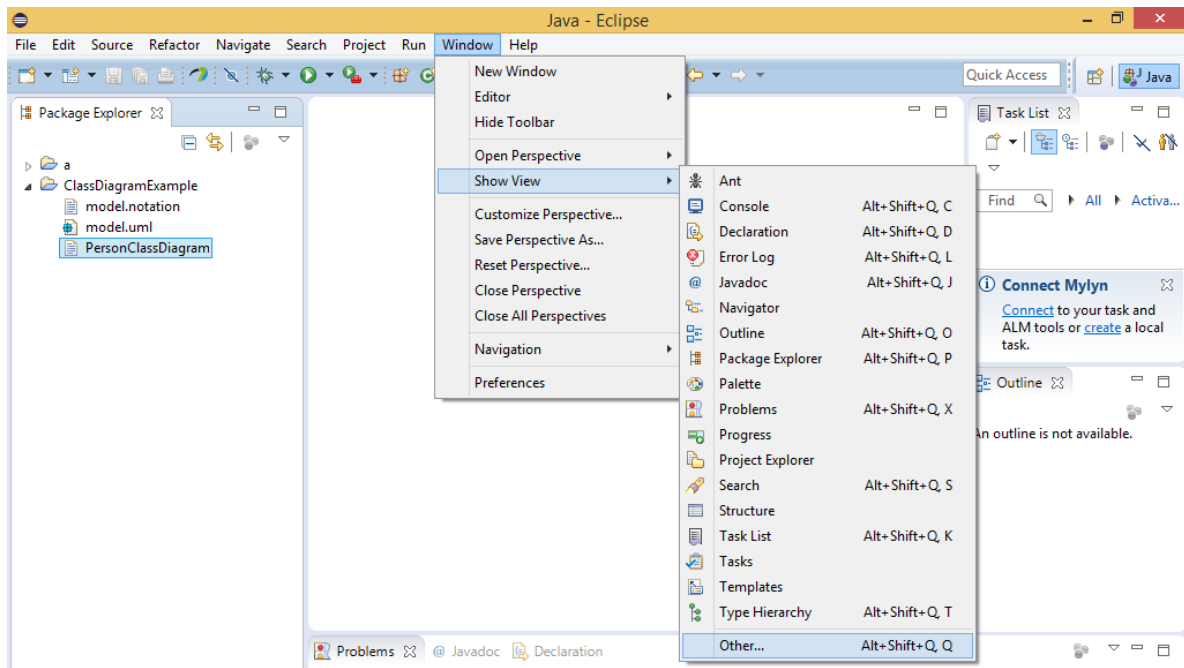
На наступному кроці потрібно обрати UML-діаграму у вікні New Papyrus Project (Select Language of the diagram): обираємо тип UML Class Diagram та завершуємо ->Finish.



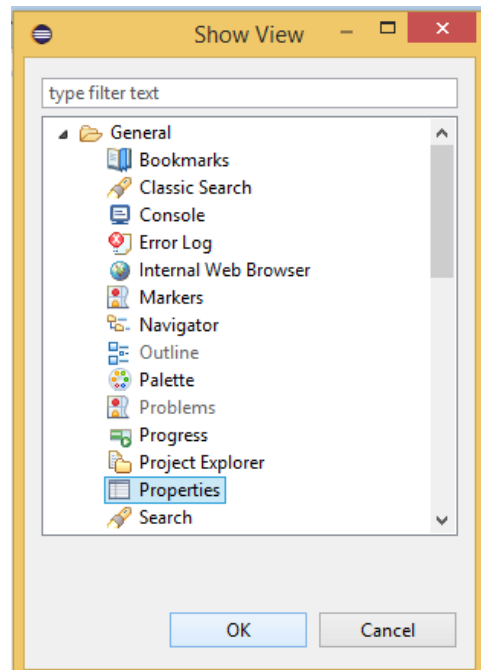
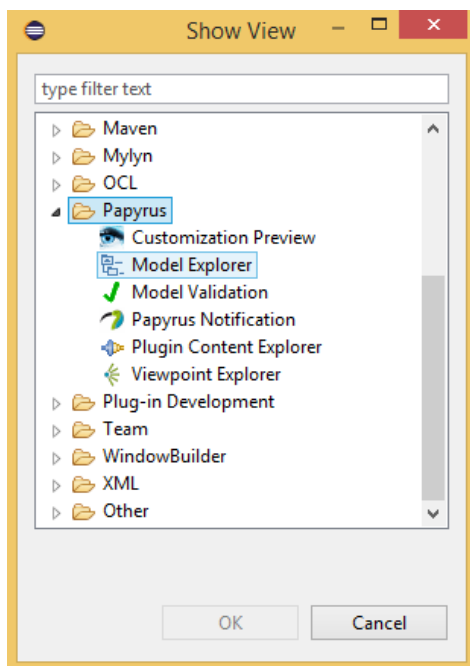
Наступний крок: у Project Explorer правим кліком по model обираємо Rename->"PersonClassDiagram.di"



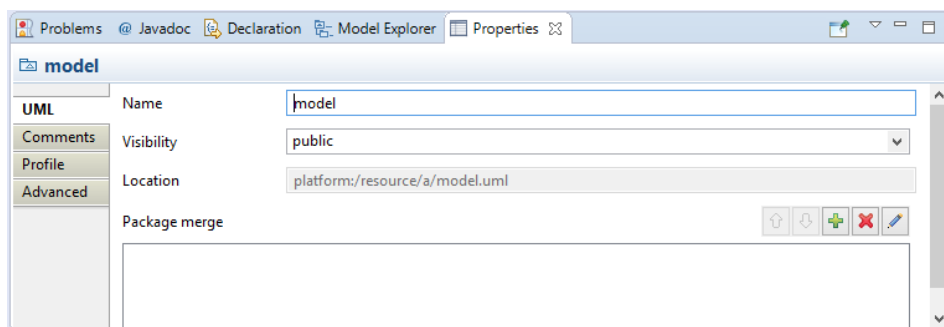
Далі нам необхідно вивести на нижню панель Model Explorer та Properties (якщо їх нема у робочому просторі за замовчуванням). Заходимо в меню Window->Show View ->Other...



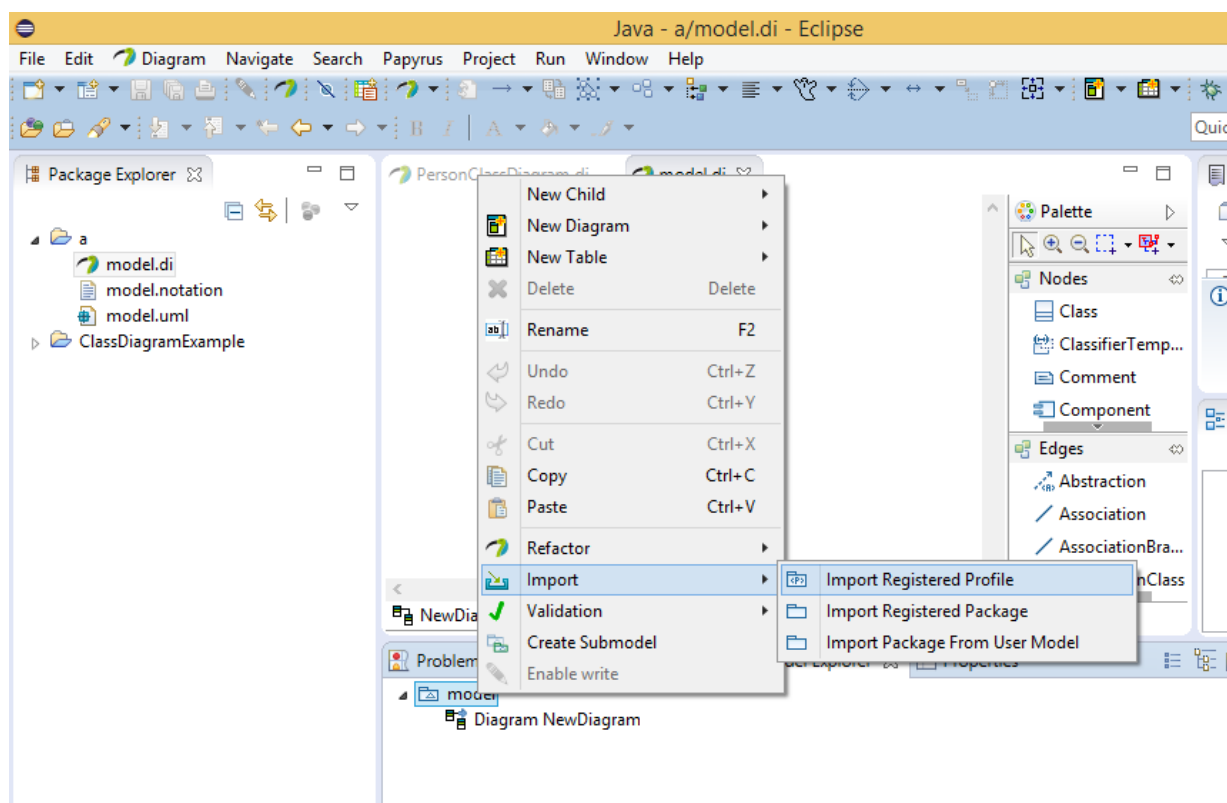
Обираємо Papyrus->Model Explorer та у General->Properties:



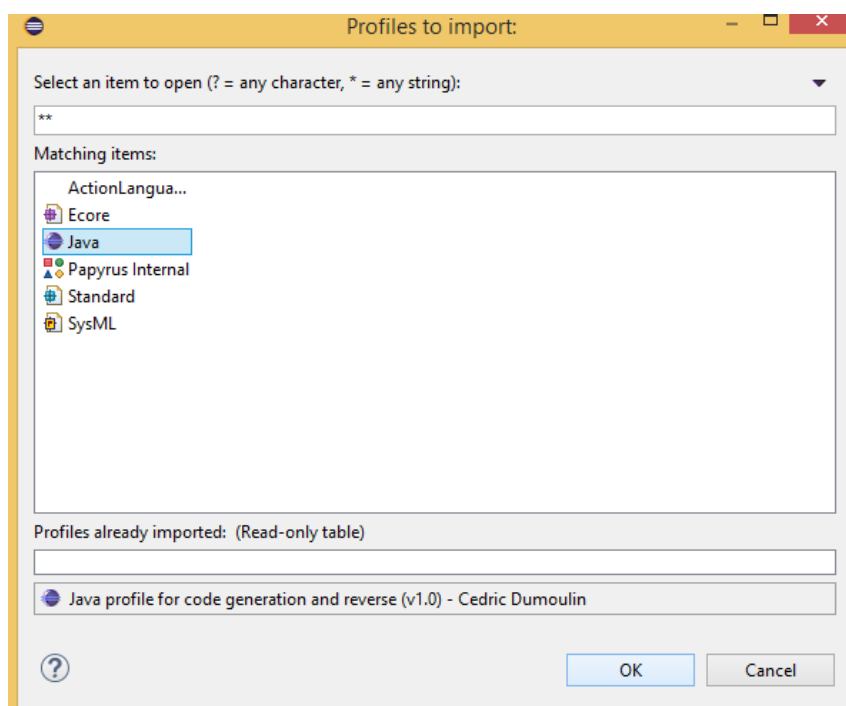
В результаті нижня панель Eclipse буде мати наступний вигляд:



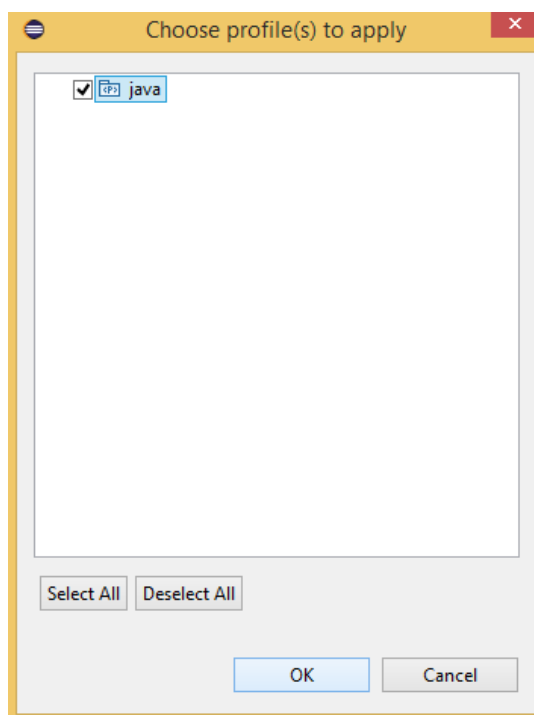
Наступний крок - у вікні Model Explorer правим кліком по model-обираємо Import->Import Registered Profile:



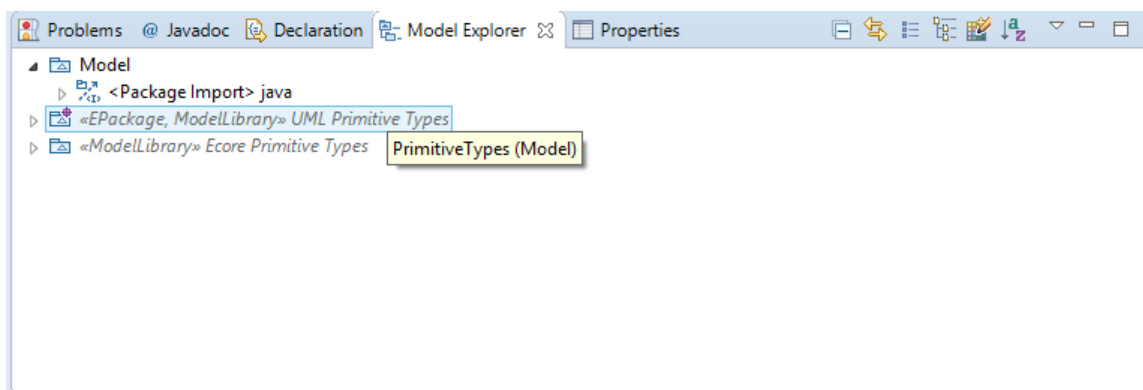
та обираємо у Profiles to import->Java



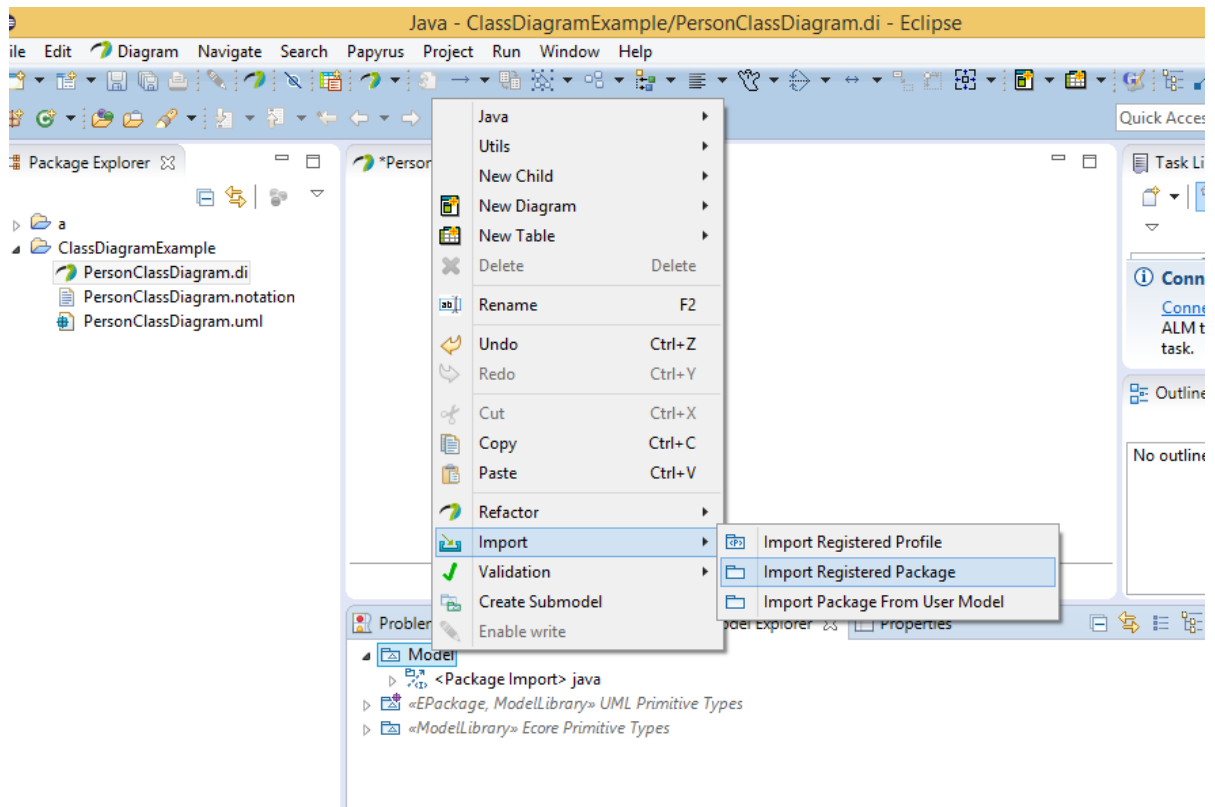
Після підтвердження з'явиться вікно в якому ставимо галочку



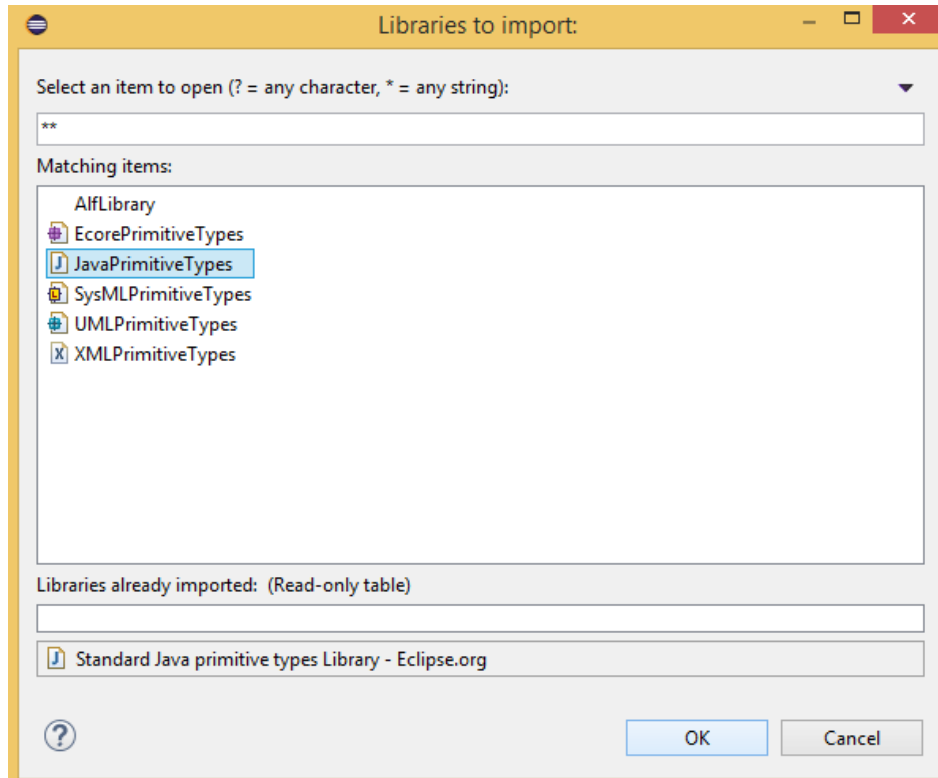
В результаті у закладці Model Explorer буде відображено наступне

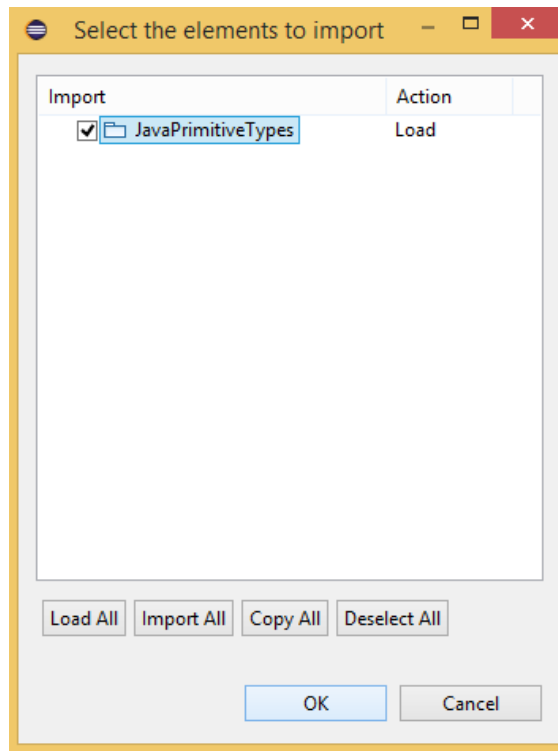


У вікні Model Explorer правим кліком по model- обираємо Import->Import Registered Package

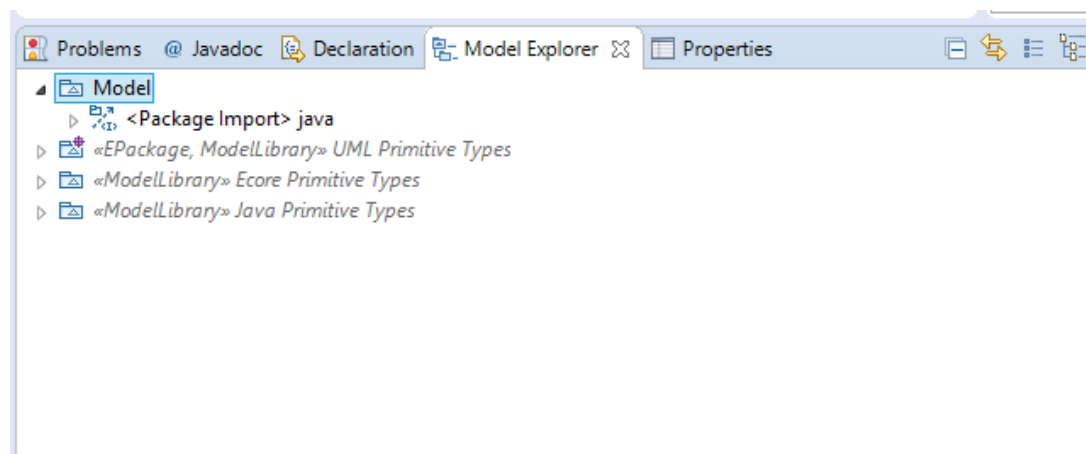


де потрібно обрати бібліотеку Java Primitive Types так як надалі будемо її використовувати

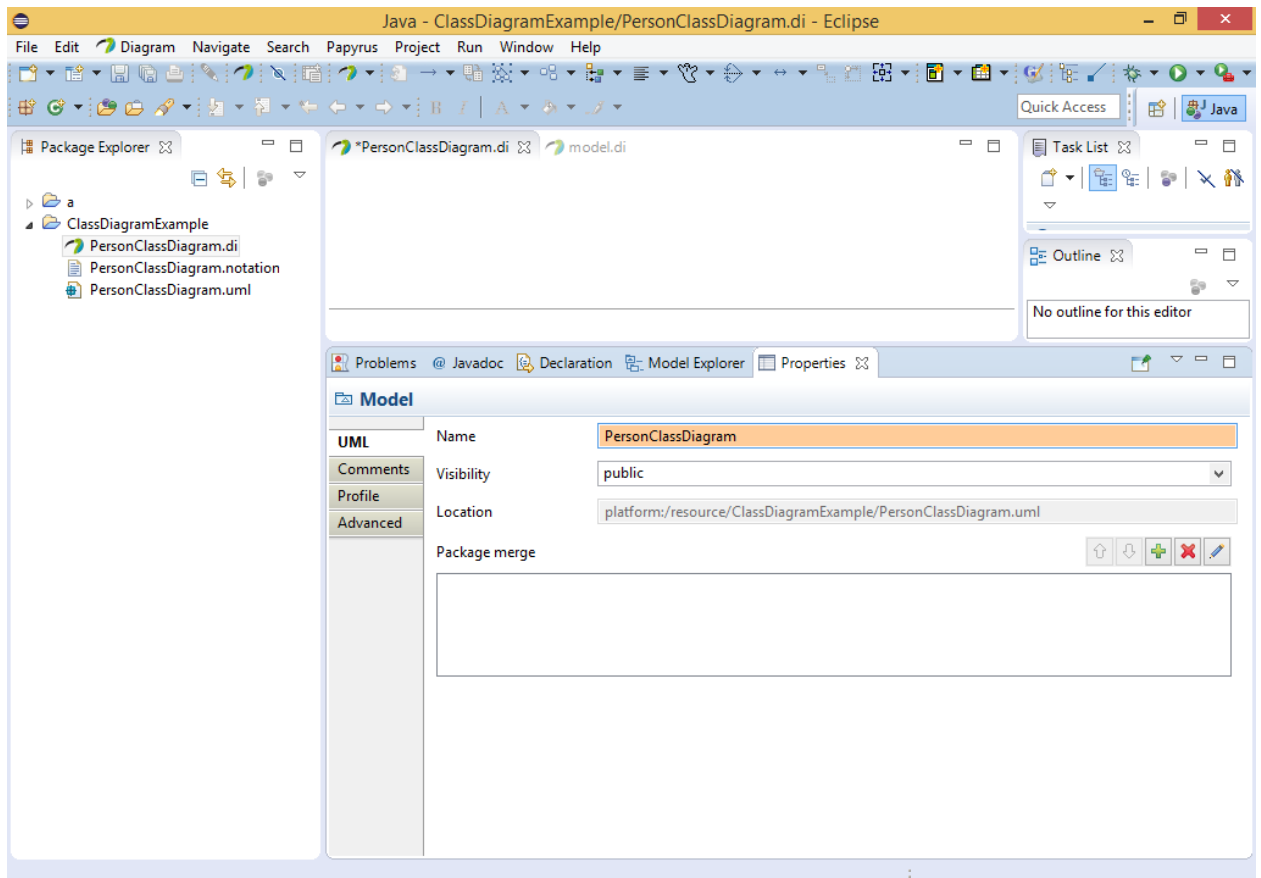




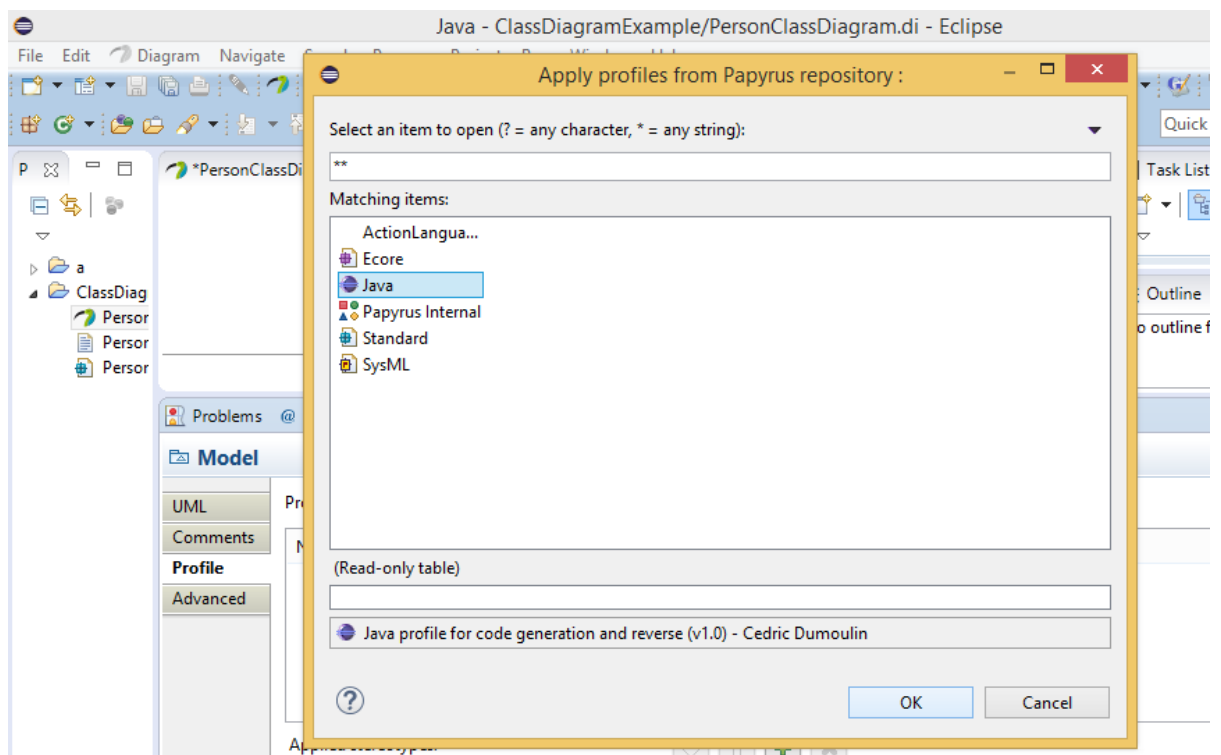
В результаті у Model Explorer з'явиться наступний запис



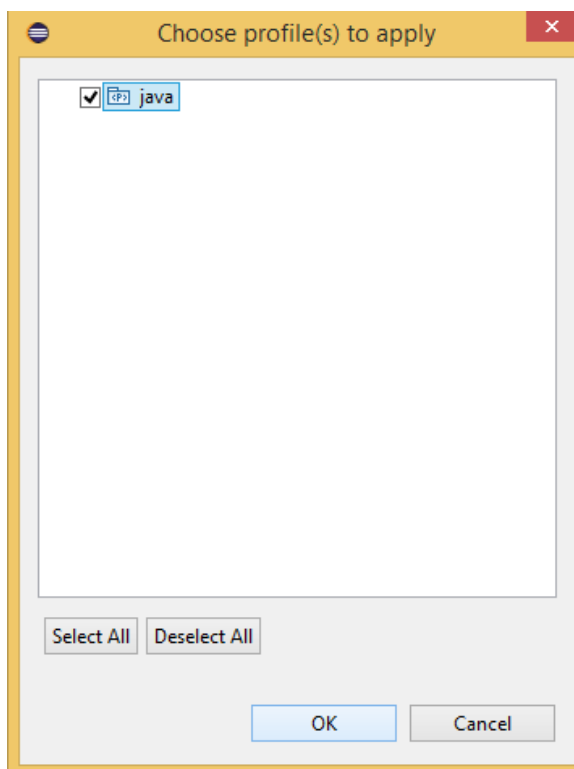
Далі переходимо на закладку Properties для нашої моделі де задаємо їй ім'я (наприклад PersonClassDiagram)



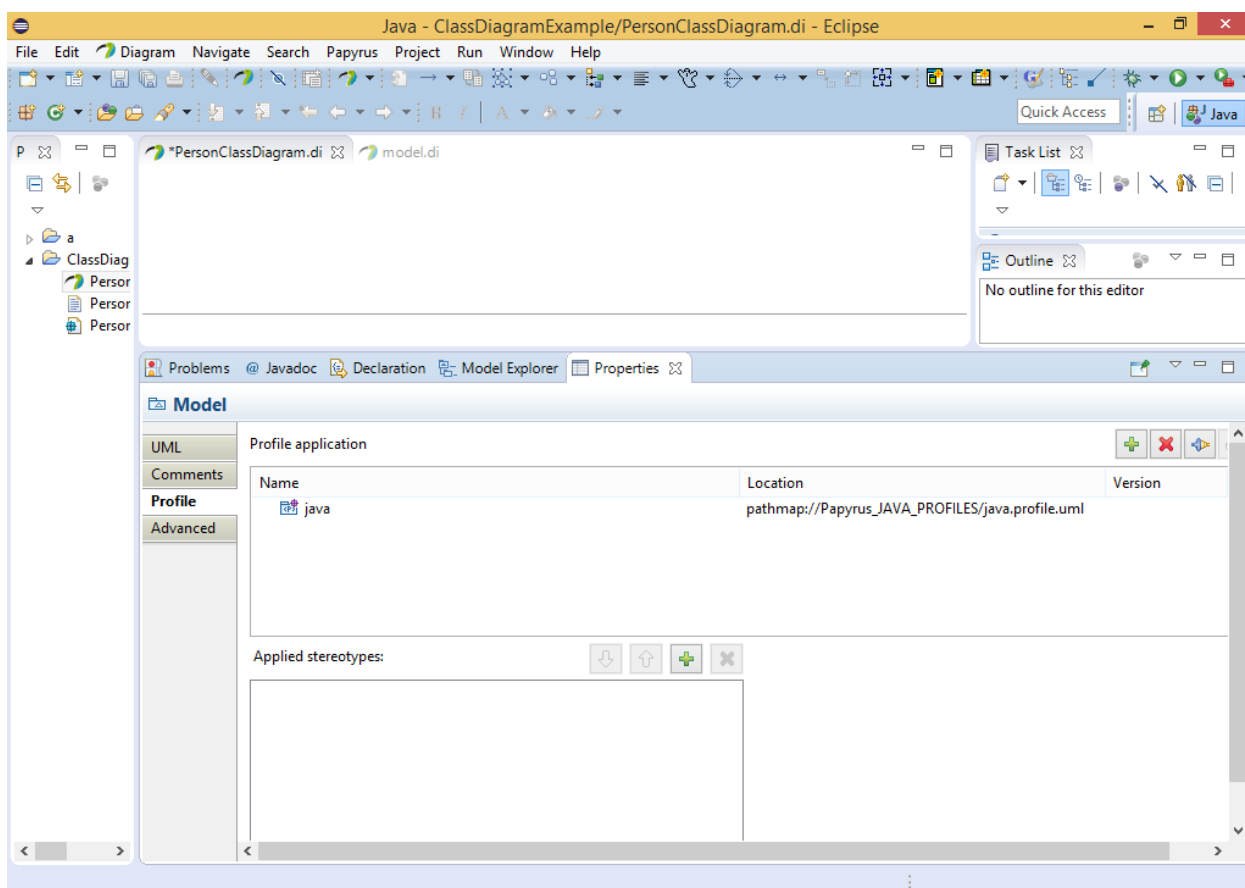
На пункті Profile (у Properties) обираємо кнопку Apply Registered Profile, у відкрившому вікні потрібно обрати Java:



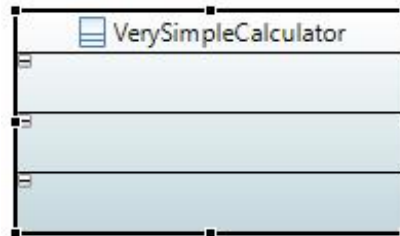
Підтверджуємо Java у Choose profile(s) apply



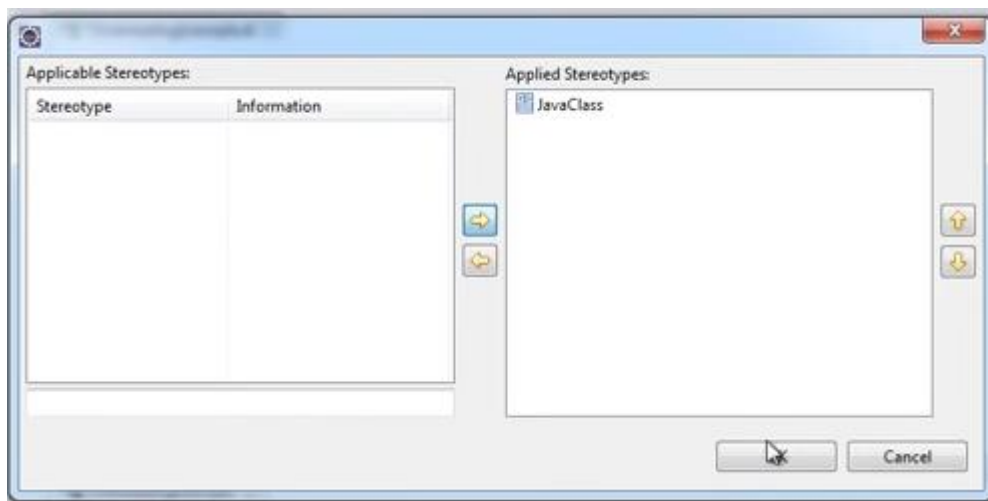
В результаті вікно Properties->Profiles приймає наступний вигляд



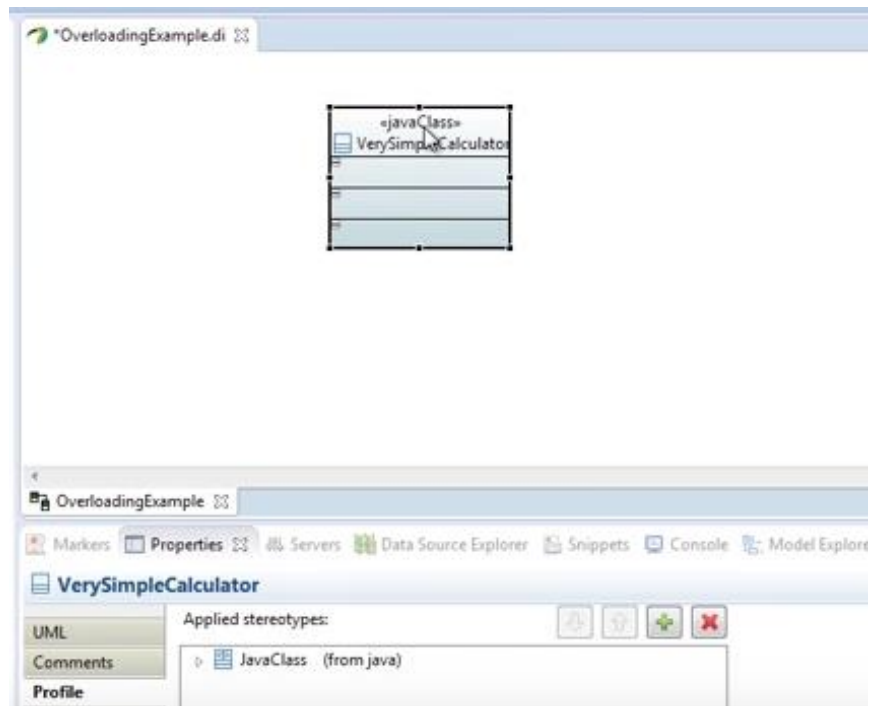
Для початку розглянемо приклад побудову моделі класу з перевантаженими методами в плагіні Papyrus. Для чого на початку оголосимо простий клас `VerySimpleCalculator`



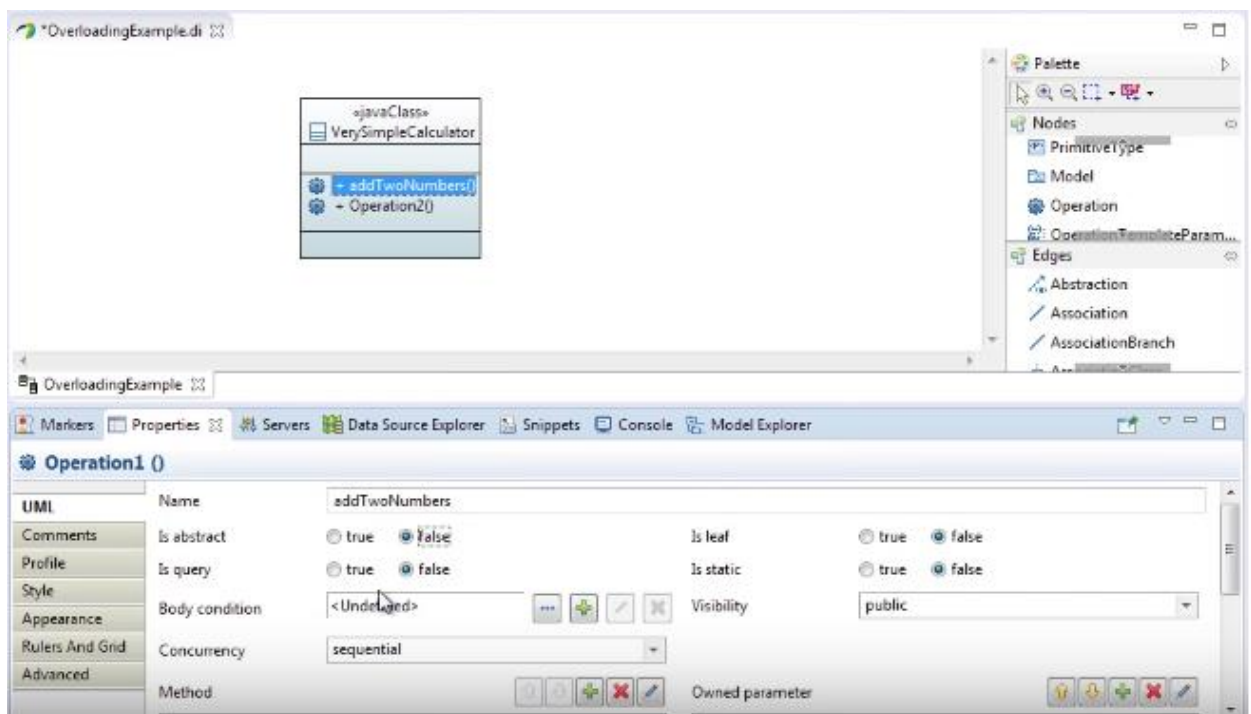
для якого в закладці `Properties`, вибіраємо `Profile-> JavaClass (Added Stereotypes)`



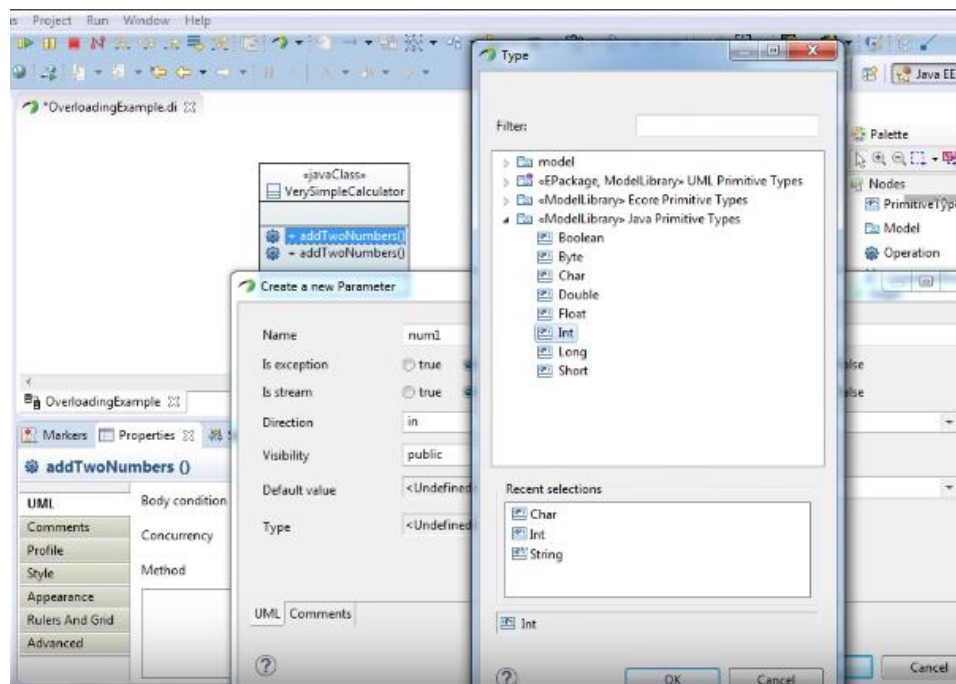
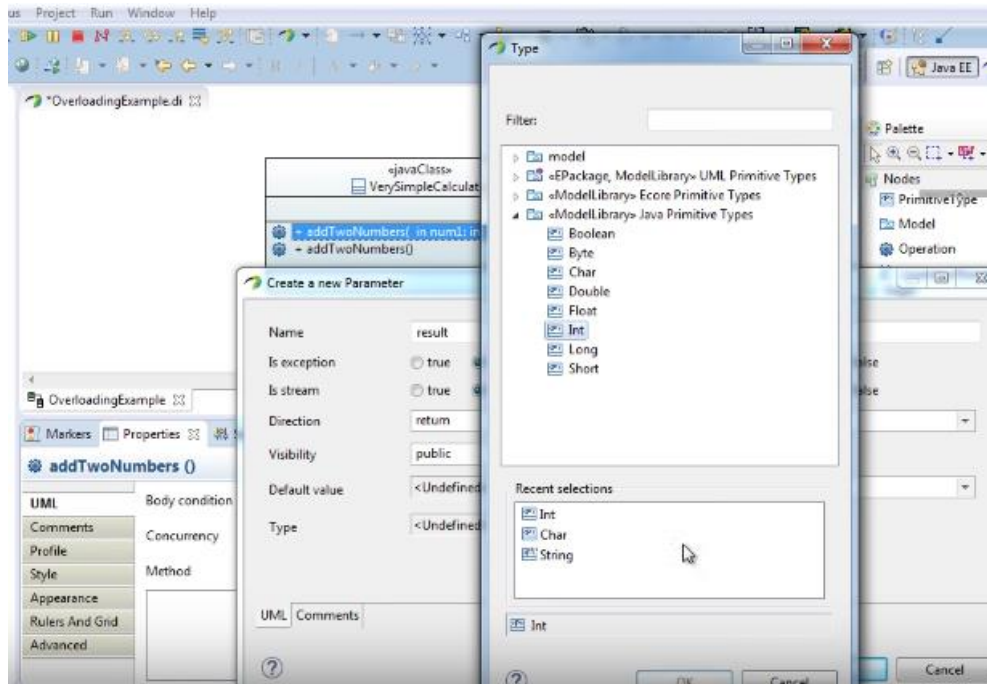
Підтверджуємо вибір, після чого маємо:



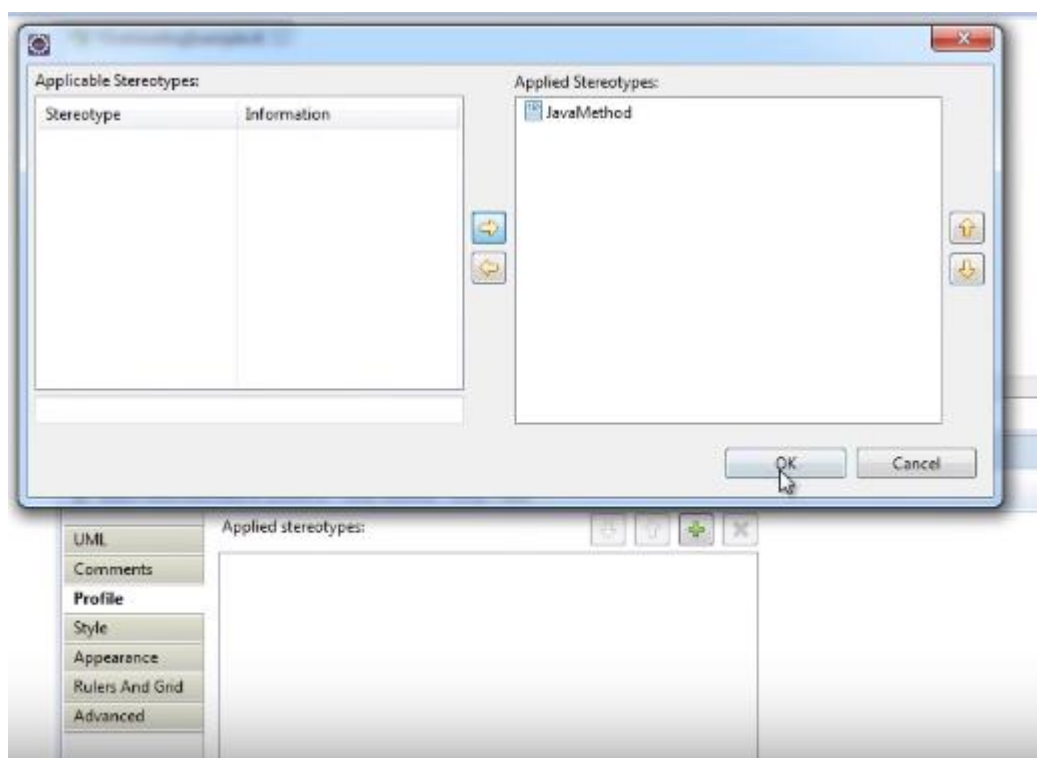
Далі додамо в наш клас кілька методів: addTwoNumbers - методи з однаковими іменами але різним набором параметрів.



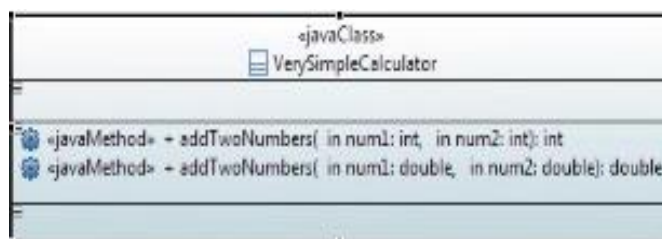
Додамо для першого методу такі параметри: (Direction-> in) - num1: Integer, num2: Integer, а також один вихідний параметр - результат (result). Для нього Direction-> return.



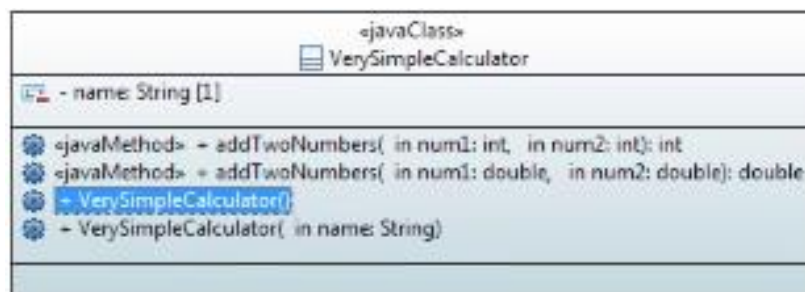
Після опису методів необхідно вказати в Profile-> Stereotypes = JavaMethods, для кожного з них.



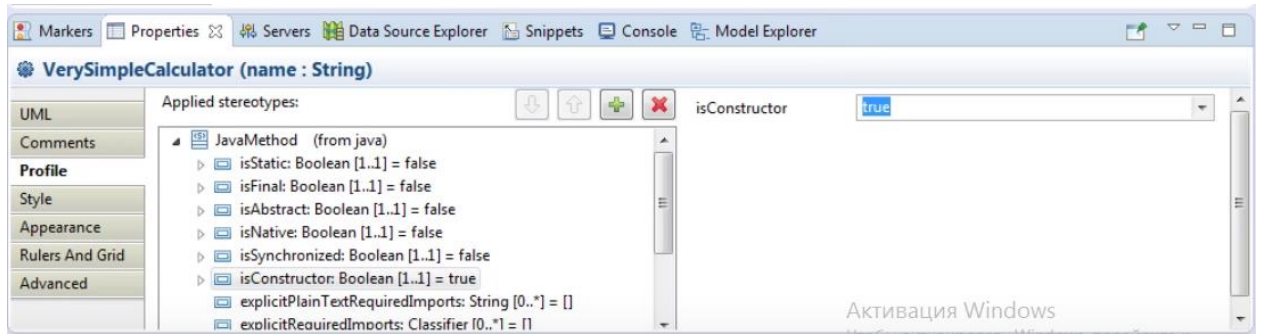
В результаті маємо клас з одним полем name і двома методами.



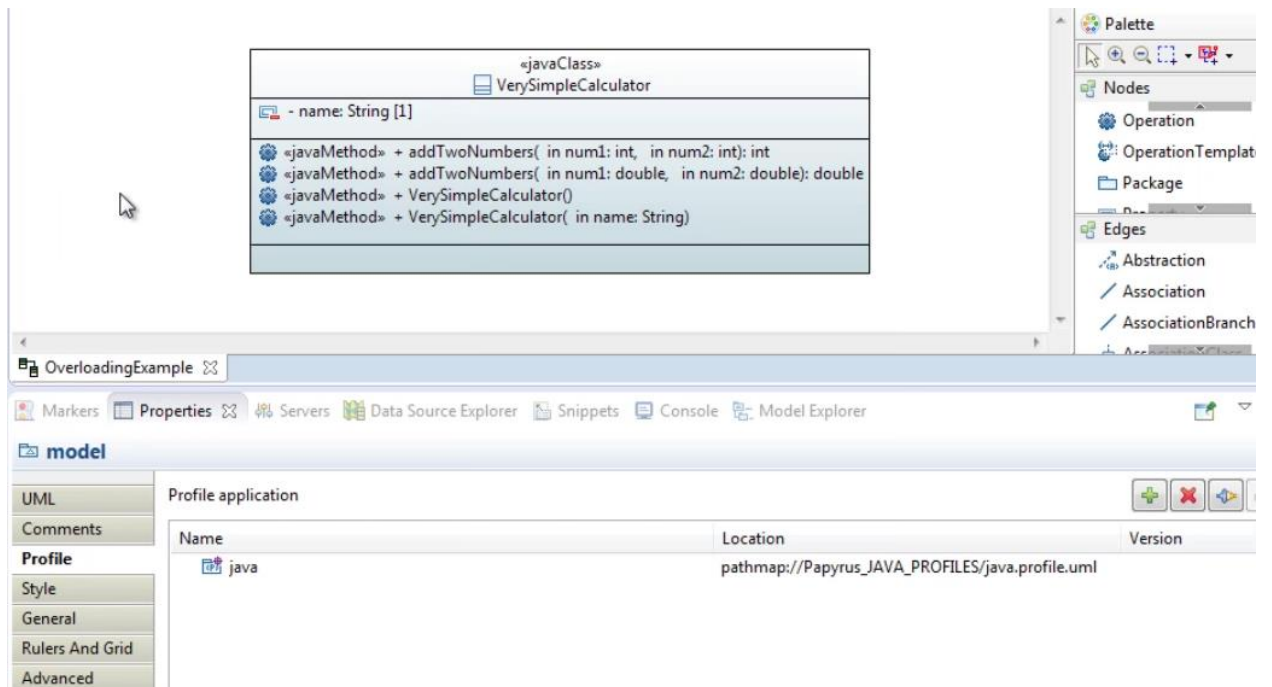
Тепер додабим нашому класу два конструктора, імена яких повинні збігатися з ім'ям класу. Отже, для першого конструктора додамо параметр name: String (EDataType).



Для кожного конструктора в Profile-> JavaMethod, і isConstructor-> true

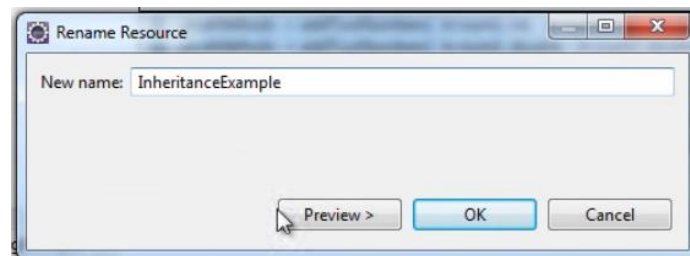
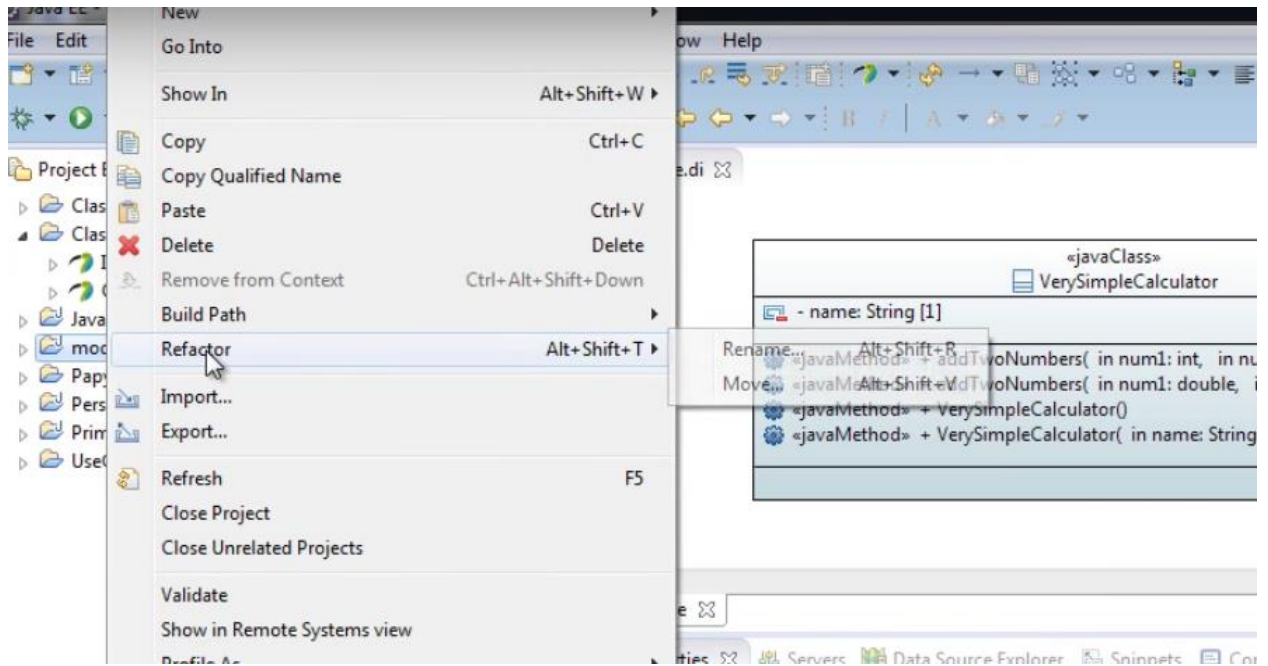


В результаті у нашого класу оголошено 2 методу і 2 конструктора.

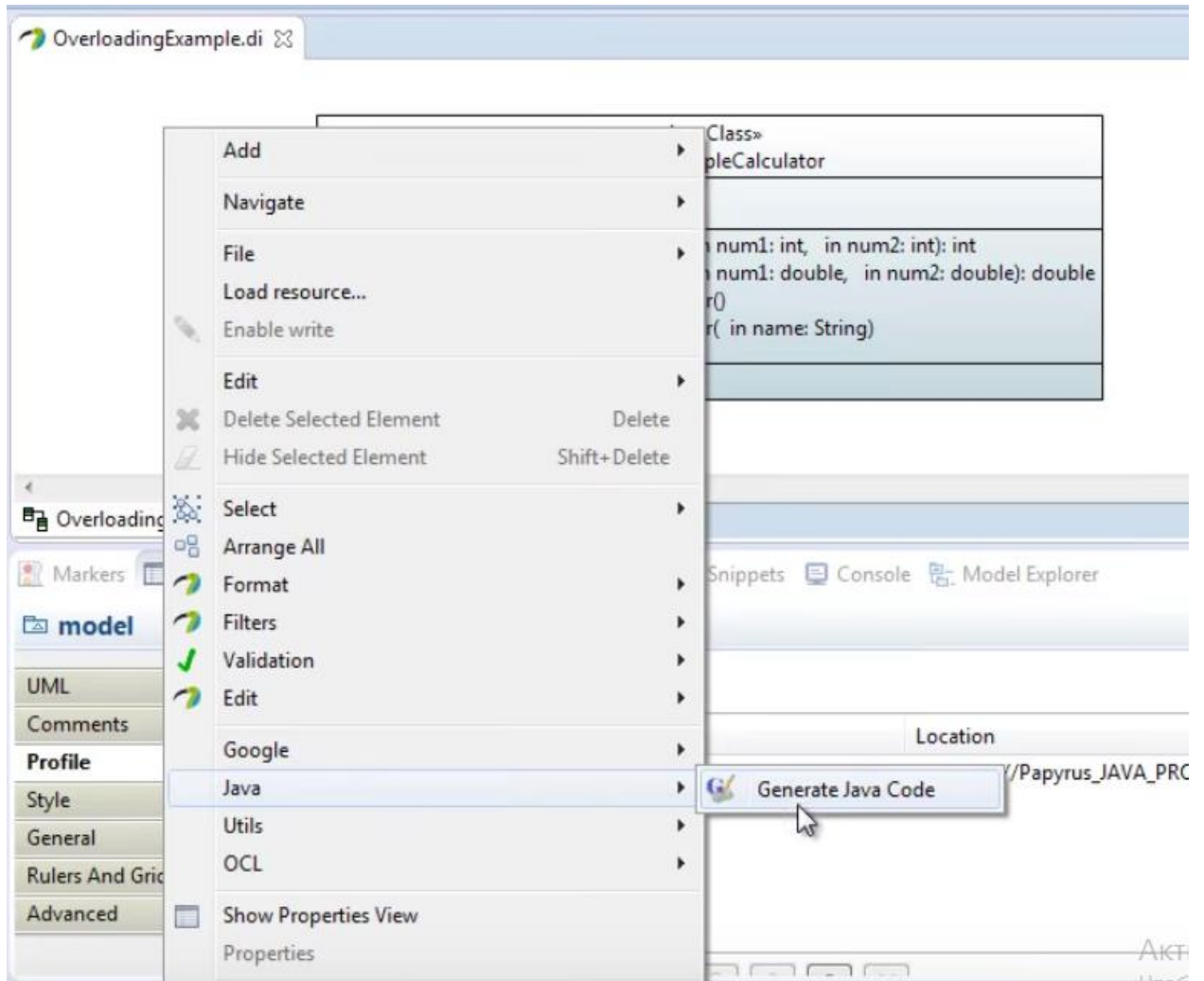


На слід. Етапі сгенерируем Java - код:

1 в Project Explorer вибираємо model- правим кліком Refactor-Rename



Тепер можна згенерувати Java- код. Правим кліком клацаємо по діаграмі класу та обираємо опцію Java – Generated Java Code



Лістинг нашої програми має наступний вигляд:

```

public class VerySimpleCalculator {

    /**
     *
     */
    public String name;

    /**
     * Getter of name
     */
    public String getName() {
        return name;
    }

    /**
     * Setter of name
     */
    public void setName(String name) {
        this.name = name;
    }
}

```

```

/**
 *
 */
public void setName(String name){

    this.name=name;
}

/**
 *
 * @param num1
 * @param num2
 */
public int addTwoNumbers (int num1, int num2){
    //TODO Auto-generated method
    return 0;
}

/**
 * @param num1
 * @param num2
 */
public int addTwoNumbers (double num1, double num2){
    //TODO Auto-generated method
    return 0;
}

/**
 *
 */
    public VerySimpleCalculator (){
        //TODO Auto-generated method

        /**
 *
 */

public VerySimpleCalculator (String name){
    //TODO Auto-generated method

```

Далі допишемо деякі доповнення до отриманного коду:

```

public VerySimpleCalculator (){
    this.name = null;
}

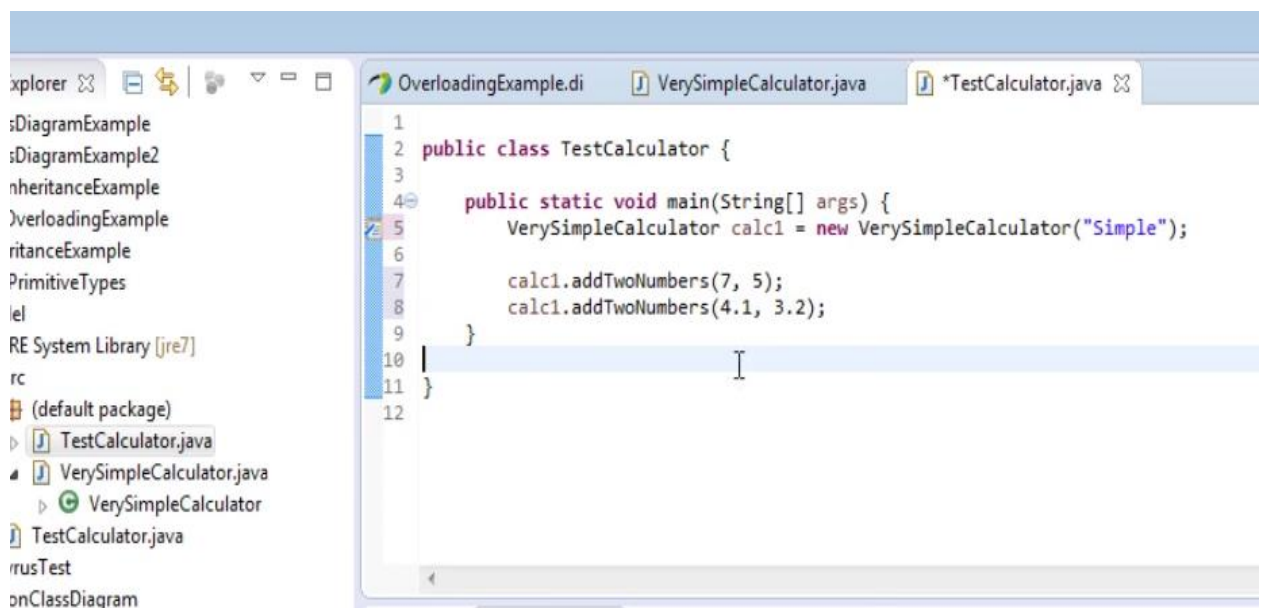
public VerySimpleCalculator (String name){
    this.name = name;
}

```

```
public int addTwoNumbers (int num1, int num2) {  
  
    return num1+num2;  
}
```

```
public int addTwoNumbers (double num1, double num2) {  
  
    return num1+num2;  
}
```

Створемо клас для тестування VerySimpleCalculator



Постановка завдання та вихідні дані для варіантів:

Завдання:

1. Опишіть клас fillSeries, що використовується для створення об'єктів з полями, що заповнюються миттєвими значеннями деякої послідовності, що генерується і має в своєму складі наступні елементи:

- набір полів, що входять до складу кожного створюваного представника даного класу;
- набір полів класу, які використовуються, як параметри послідовності, що генерується;
- блок ініціалізації, в якому відбувається ініціалізація полів об'єктів;
- метод, який виводить на екран інформацію про поля даного об'єкта;
- метод, який повертає характеристику об'єкта - значення деякої функції, в обчисленні якої використовуються поля даного об'єкта;
- інші методи, необхідні в залежності від специфіки варіанту завдання;

2. У методі main оголошіть необхідні масиви, проініціалізуйте їх елементи та виконайте необхідні дії для вирішення конкретного завдання в залежності від варіанту. Повторювані послідовності операторів можна оформити у вигляді статичних методів, виходячи із специфіки задачі. Хід вирішення повинен супроводжуватися наочним виводом на екран інформації, що дозволяє контролювати його кроки.

Варіант 1

У цеху працюють три технологічні лінії по розливу фарби в банки з наступним алгоритмом роботи: кожні "interval" годин автомат ллє фарбу одного кольору, після чого переходить на наступну. Фарби чергуються в такому порядку: "червона", "зелена", "жовта", "синя", "біла" і "блакитна". Маса фарби в кожній банці повинна бути за нормою "massnorm" кг, але коливатися в діапазоні "diarazon" (+ / -) 25 грам для 1-го автомата, (+ / -) 10 грам - для 2-го і (+ / -) 50 грам - для 3-го. Продуктивність всіх ліній 200 банок на годину.

Промодельуйте ситуацію роботи верстатів в наступному режимі:

- 1-й почав роботу о 10 годині з "зеленого" кольору і пропрацював 36 годин з інтервалами розливу 2:00;
- 2-й почав роботу в 0 годин з "білого" кольору і пропрацював 45 годин з інтервалами розливу 5:00;
- 3-й почав роботу в 20 годин з "жовтого" кольору і пропрацював 25 годин з інтервалами розливу 4:00; Визначте:
- скільки білої фарби розлили три лінії разом?
- яка з ліній розлила більше блакитної фарби?
- наскільки відхилилася реальна маса фарби розлитої 2 лінією за час її роботи від зазначеної за нормами при даній кількості банок?

Варіант 2

На лабораторному стенді стоять два стрічкових самописця, що фіксують вихідний параметр дослідних агрегатів кожну хвилину. Кожен агрегат видає якийсь синусоїдальний сигнал виду " $\sin(t)$ ". На цей сигнал накладаються випадкові флуктуації, що спотворюють сигнал на 12% для 1-го і на 8% для другого агрегату, зважаючи на їх недосконалість. Ще один самописець контролює ідеальний агрегат, похибка якого 0%.

Проімітуйте роботу даного стенду на протязі 30 хв і визначте:

- в які хвилини 1-й самописець фіксував сигнал, що перевищує крайні амплітудні значення щодо ідеального сигналу?
- скільки було аналогічних ситуацій у 2-го приладу?
- яка середня величина відхилень для першої половини інтервалу вимірювання у 1-го самописця і для другої половини інтервалу вимірювання для 2-го самописця?

Варіант 3

Врожайність висадженого насіння деякої тепличної культури залежить від пори року таким чином: зима - 70%, весна - 95%, літо - 80%, осінь - 65%. У

трьох теплицях протягом року вирощувалася дана культура: в 1-й теплиці - 1200 штук насіння на місяць, аналогічно в 2-й - 2700 і в 3-й - 2100. Через різну якість ґрунту середня вага одного плоду для кожної теплиці складає 2,2, 3,8 і 1,6 кг відповідно. Повний термін дозрівання 1 місяць для всіх.

Промоделюйте ситуацію і визначте:

- яка теплиця дала найбільший урожай за весь рік?
- порівняйте "літньо-осінній" врожай першого теплиці і "зимововесняний" врожай третьої теплиці?
- який урожай був зібраний з трьох теплиць влітку?

Варіант 4

Барабан центрифуги розганяється до робочої частоти " N_{\max} " (об/с) з деяким прискоренням " a " від деякої початкової частоти " n_0 ". Миттєва частота обертання обчислюється за формулою $N = n_0 + a \cdot t$. Три барабани з радіусами 50, 75 і 100 см були розігнані від початкової частоти 0, 20 і 90 об/с до робочої 300 об/с з прискоренням 10 об/с². Лінійна швидкість обчислюється за формулою " $V = 2 \cdot \pi \cdot R \cdot N$ ".

Промоделюйте процеси розгону кожного і визначте:

- який барабан розганявся довше за всіх і наскільки?
- в який момент часу швидкість 2-го барабана дорівнювала половині N_{\max} ?
- яка була лінійна швидкість двох ще не розігнаних барабанів на момент кінця розгону найшвидшого?

Варіант 5

У бункер-змішувач насипається три компоненти майбутньої суміші, кожен компонент по своїй трубі. Засувки кожної труби працюють в режимі

відкрита/закрита з рівними інтервалами часу. Кожну секунду 1-а труба насипає 35 см³, 2-га - 70 см³ і 3-я - 24 см³ компонента.

Промоделюйте ситуацію, коли труби працювали 1 хв з інтервалами 35, 26 і 18 с і насипали пісок, щебінь і цемент. Визначте:

- якої за обсягом суміші в бункер було засипано більше і наскільки?
- скільки кг цементу насипала третя труба?
- скільки часу треба було працювати трубам, щоб маса насипаного кожною трубою компонента була б однаковою?

Варіант 6

З деякої висоти в горизонтальному напрямку з деякою початковою швидкістю « v_0 » вистрілює снаряд. Стробоскопічний пристрій фіксує висоту кожно 0,1 секунду польоту, закінчуючи роботу при падінні снаряда на землю. Дальність польоту при цьому можна визначити, як $S = v_0 * T_{\max}$. Було випущено 3 снаряди з висоти 100, 350 і 560 метрів з початковою швидкістю 430, 200 і 145 м/с відповідно.

Визначте:

- скільки часу тривав політ кожного снаряду?
- яка відстань між точками падіння найдалшого і найближчого снаряду?
- на якій висоті знаходилися снаряди, які залишилися в повітрі в момент падіння першого з них?

Варіант 7

Вантажний залізничний потяг формується таким чином: перші "N" вагонів мають об'єм 250 м³, наступні "2*N" - 200 м³, наступні "3*N" - 150 м³, а всі інші - 100 м³. Було сформовано три потяги із 45 вагонів, кожен з параметрами: N = 5 - для 1-го, N = 3 - для 2-го і N = 8 - для 3-го. У 1-й потяг було завантажено дерево, у 2-й - вугілля, а в 3-й - пісок.

Визначте:

- який з потягів був навантажений максимально?
- скільки вагонів об'ємом 100 м в кожному потязі?
- наскільки відрізняється навантаження вагонів, які мають об'єм 250 м³, третього потягу від суми вагонів, які мають об'єм 150 м³, 1-го і 2-го потягу?

Варіант 8

Народженим кошенятам чоловічої статі господарі дали наступні клички: "Васька", "Мурзик", "Тимофій" і "Сніжок", а кошенятам жіночої статі: "Мурка", "Пушинка", "Ліза" та "Люся", послідовно перебираючи клички в порядку їх слідування в зазначеному переліку. Перший раз кішка народила 4 кошенят, 2-й раз - 8 і 3-й раз 5, причому стать кошенят була випадковою. Одне кошеня коштує в середньому 10 грн, "Мурки, Васьки" - на 2 грн дешевше, а "Сніжки, Пушинки" - на 5 грн дорожче.

Визначте:

- кошенят якої статі більше народилося у кішки за три рази і наскільки?
- чи були "Мурзики" в 2-му поколінні і "Люсі" в 3-му і скільки?
- скільки грошей заробили господарі на 1-му і 3-му поколінні?

Варіант 9

Новорічний дитячий подарунок складається з: X гр шоколадних цукерок, $X \cdot 1,5$ гр льодяників, $X \cdot 2,3$ гр вафель, все інше - мандарини. Було сформовано 3 види подарунків з різними упаковками, вагою 800 гр кожен: $X = 150$ - для першої упаковки, $X = 125$ - для другої і $X = 130$ - для третьої. Вартість шоколадних цукерок – 45 грн / кг, льодяників – 22 грн / кг, вафель - 27 грн / кг, мандарин - 14 грн / кг.

Визначте:

- вагу кожної складової в кожному подарунку?
- ціну кожного подарунка?
- в якому подарунку шоколадних цукерок було більше?

Варіант 10

У музей приїхала нова експозиція. Вхідний квиток для дорослих коштує 60 грн, знижка для студентів - 50%, а дитячий квиток на 40 грн дешевше від дорослого. У перший день музей відвідали 67 осіб, у другій - 84, а в третій - 72, причому вік людей був випадковий.

Визначте:

- скільки дітей відвідали музей за три дні?
- скільки грошей було в касі музею в перший день?
- чи були студенти в музеї у другий день і скільки?

Контрольні питання

1. Дайте визначення об'єктно-орієнтованого програмування.
2. Поняття класу та об'єкта в ООП.
3. Поняття поля та методу в ООП.
4. Статичні та нестатичні компоненти класу.
5. Дайте визначення блоку ініціалізації.

ЛАБОРАТОРНА РОБОТА №2

«Створення UML- діаграми для роботи з конструкторами класів.

Перевантаження методів»

Мета: Ознайомитися з принципами роботи конструкторів, навчитися використовувати їх для різних варіантів початкової ініціалізації полів об'єктів. Вміти будувати класи, які використовують перевантажені методи, і застосовувати їх для вирішення практичних завдань.

Рекомендації до виконання роботи (частина 1):

Ініціалізація полів конструкторами

Часто виникає необхідність керувати процесом створення представника класу, щоб отримувати поля, заповнені тим чи іншим чином, залежно від вимоги конкретної ситуації.

Наприклад, при створенні класу "коробка сірників", можна прийняти, що середньостатистична коробка має наступні параметрами: "кількість" - 100 штук, "довжина" - 5 см, "матеріал" - осика, "колір наконечника" - коричневий, "виробник" - у загальному випадку неясний, тому його можна задати як порожній рядок " " - за даним шаблоном можуть створюватися представники за замовчуванням.

```
class matches_package {  
    int number;  
    double length;  
    String material;  
    String producer = " ";  
  
    matches_package ( ) {  
        number = 100;  
    }  
}
```

```

    length = 5;
    material = "aspen";
    color = "braun";
    }
public static void main (String args []) {
    matches_package pm = new matches_package ();
}
}

```

У даному випадку представник `pm` і всі інші, створені подібним чином, будуть мати поля, задані конструктором за замовчуванням. При подальшій роботі значення полів можна змінювати відповідно з необхідністю, але початковий їх стан при ініціалізації буде визначено конструктором.

Конструктор - це спеціальний метод, який визивається інтерпретатором Java щоразу, коли оператором `new` створюється черговий представник класу. Він носить ім'я класу і не містить вказівок щодо типу значення, що повертається.

Конструкторів може бути кілька, всі вони будуть описані як `matches_package`, але відрізнятимуться набором і типом аргументів.

Припустимо, є необхідність створювати типових представників коробки з сірниками, але відразу вказуючи їх кількість в коробці. Для цього можна або скористатися описаним раніше конструктором за замовчуванням, а після цього модифікувати поле `number`, або оформити ці дії у вигляді ще одного конструктора, який буде викликатися при наявності аргументу цілого типу при створенні об'єкта, наприклад: `new matches_package (50)`.

```

matches_package (int number) {
    this ();
    this.number = number;
}

```

У цьому випадку в першу чергу буде викликаний конструктор без аргументів `this ()`, а потім полю `number` буде присвоєно передане значення.

Слово `this` - це вказівка на даного представника класу, тому рядок `this.number = number` слід розуміти так: присвоїти полю `number` даного об'єкта, значення локальної змінної `number`, яке передано у вигляді аргументу. У цьому випадку уникаємо "конфлікту імен". Якщо ім'я локальної змінної в аргументі конструктора не співпадає з ім'ям поля, то `this` можна не вказувати - це буде надмірність, хоча і з нею все буде працювати правильно.

У класі не може бути двох конструкторів з однаковим набором аргументів. Наприклад, якщо по-аналогії необхідно створювати об'єкти в 1му випадку - зі відразу зазначеним матеріалом, а в 2-му - з вказаним кольором наконечника, то компілятор не дозволить цього зробити, оскільки і "матеріал" і "колір" мають тип `String` і при ініціалізації об'єкта командою:

```
new matches_package ("Red")
```

було б незрозуміло, якому полю типу `String` необхідно присвоїти значення "Red". Дану проблему можна обійти, додавши конструктор вигляду:

```
matches_package (String material, String color) {
    this ();
    this.material = material;
    this.color = color;
}
```

У цьому випадку необхідно пам'ятати, який з аргументів має бути на 1-у, а який на 2-у місці.

За допомогою конструктора можна організувати також процес створення об'єкта-копії, для цього конструктору необхідно передати об'єкт такого ж типу, як створюваний, і привласнити значення кожного його поля однойменним полям створюваного об'єкта:


```

matches_package (matches_package mp) {
    number = mp.number;
    length = mp.length;
    material = mp.material;
    color = mp.color;
}

```

Перевантаження методів

Наявність у класі декількох конструкторів - методів з однаковим ім'ям, але різним набором аргументів, є прикладом перевантаження методів. Аналогічно можна поступити наприклад в ситуації реалізації методу "порівняння", який залежно від складу аргументів буде виконувати ті чи інші дії, відмінні для кожної реалізації.

Наприклад в ситуації з сірниками, можна запропонувати кілька варіантів цього методу:

```

boolean compare () {
    if (number != 0) return true;
    else return false;
}

int compare (int number) {
    if (this.number == number)
        System.out.println ("Equale length");
    else
        System.out.println ("Not equale length");
    return
        this.number-number;
}

void compare (matches_package mp) {
    boolean temp;
    temp = (number == mp.number && length == mp.lengths &&
material == mp.material && color == mp.color)
    if (temp == true)

```

```

        System.out.println ("This objects is full equale");
    else
        System.out.println ("This objects is differenses");}

```

Перша реалізація методу не має аргументів і порівнює об'єкт сам з собою на предмет свого порожнього змісту і повертає логічне значення, друга - приймає число цілого типу та порівнює з ним кількість сірників в об'єкті, для якого цей метод викликаний, і повертає різницю, а третя - приймає об'єкт такого ж типу і перевіряє його і даний на їх повну тотожність один одному, при цьому нічого не повертаючи.

Передача масиву в якості аргумента методу

Аргументами методів можуть бути не тільки величини простих типів, а й масиви. Наприклад, ускладнимо модель, представивши довжину кожного сірника у вигляді масиву типу `double`, при довжині масиву `number` і опишемо конструктор, який буде присвоювати елементам масиву об'єкта значення елементів іншого масиву, переданого конструктору як аргумент:

```

class matches_package {
    int number;
    double length;
    double len [ ]; // додано поле len у вигляді масиву
чисел типу double
matches_package (double arr) {
    number = arr.length;
    for (int i = 0; i<number; i + +) {
        len [i] = arr [i];
    }
}
}

```

У методі `main` виклик такого конструктора може бути виконаний різними способами:

- шляхом попереднього створення тимчасового масиву "вручну":

```
double tempArr [ ] = new double [ ] {8.2, 8.1, 7.9, 7.1,
7.0, 6.9};
matches_package mp1 = new matches_package (tempArr);
```

- задавши елементи масиву в циклі (наприклад випадково):

```
double tempArr [ ] = new double [5];
for (int i=0; i <tempArr.length; i + +) {
    tempArr [i] = Math.random ( ) * 2 +6;
}
```

- за допомогою анонімного масиву:

```
matches_package mp2 = new matches_package (new double [ ]
{8.2, 8.1, 7.9, 7.1, 7.0, 6.9});
```

Текстовий файл *.java і файли *.class

Створювати і працювати з об'єктами класу можна не тільки в методі main цього ж класу, але і з іншого класу. Наприклад, протестувати роботу класу matches_package можна з класу test_matches, розмістивши в ньому метод main, в якому будуть знаходитися необхідні оператори. Текст обох класів можна розташувати в одному текстовому файлі з розширенням .java, при цьому система відкомпілює кожен клас в свій окремий .class файл.

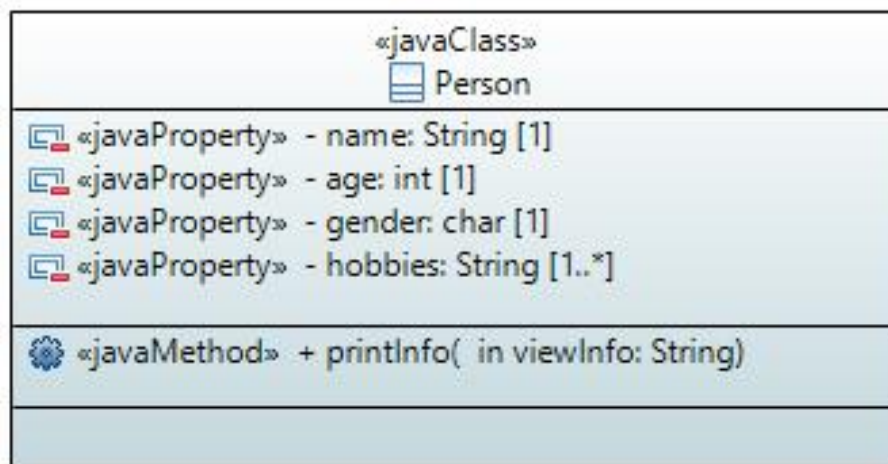
При роботі в середовищі Eclipse слід правильно задати ім'я текстового .java файлу. Оскільки налаштовані команди компіляції і запуску підставляють в командний рядок ім'я поточного файлу, то при запуску на виконання система буде шукати метод main саме в тому файлі .class ім'я якого збігається з ім'ям текстового .java файлу. Тому, у випадку з сірниками, текстовий файл, що містить класи matches_package і test_matches, слід назвати test_matches.java.

Природно, що при роботі з DOS-середовища, наприклад з використанням файлового менеджера FAR, дані обмеження не критичні,

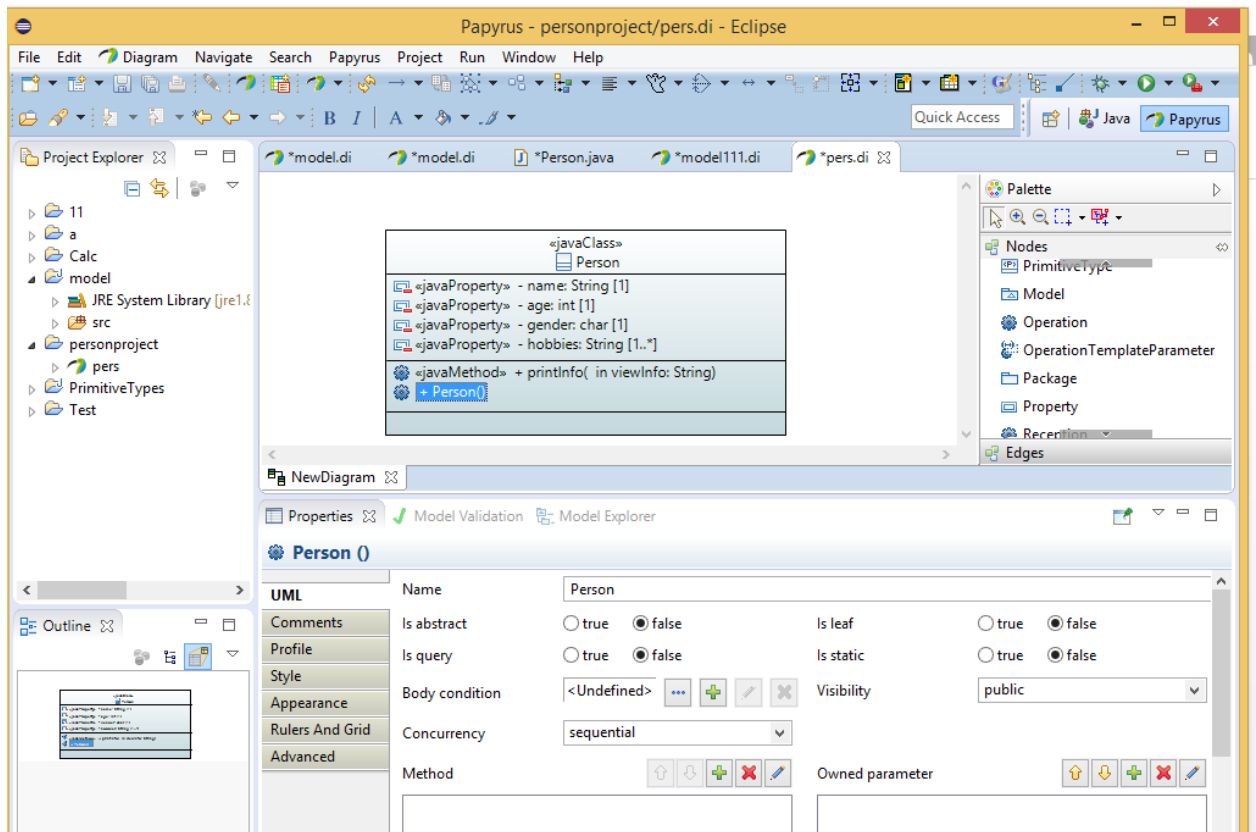
оскільки можна проводити запуск необхідного .class файла безпосередньо вказавши його ім'я в командному рядку.

Рекомендації до виконання роботи (частина 2): моделювання за допомогою плагіну Rarurys.

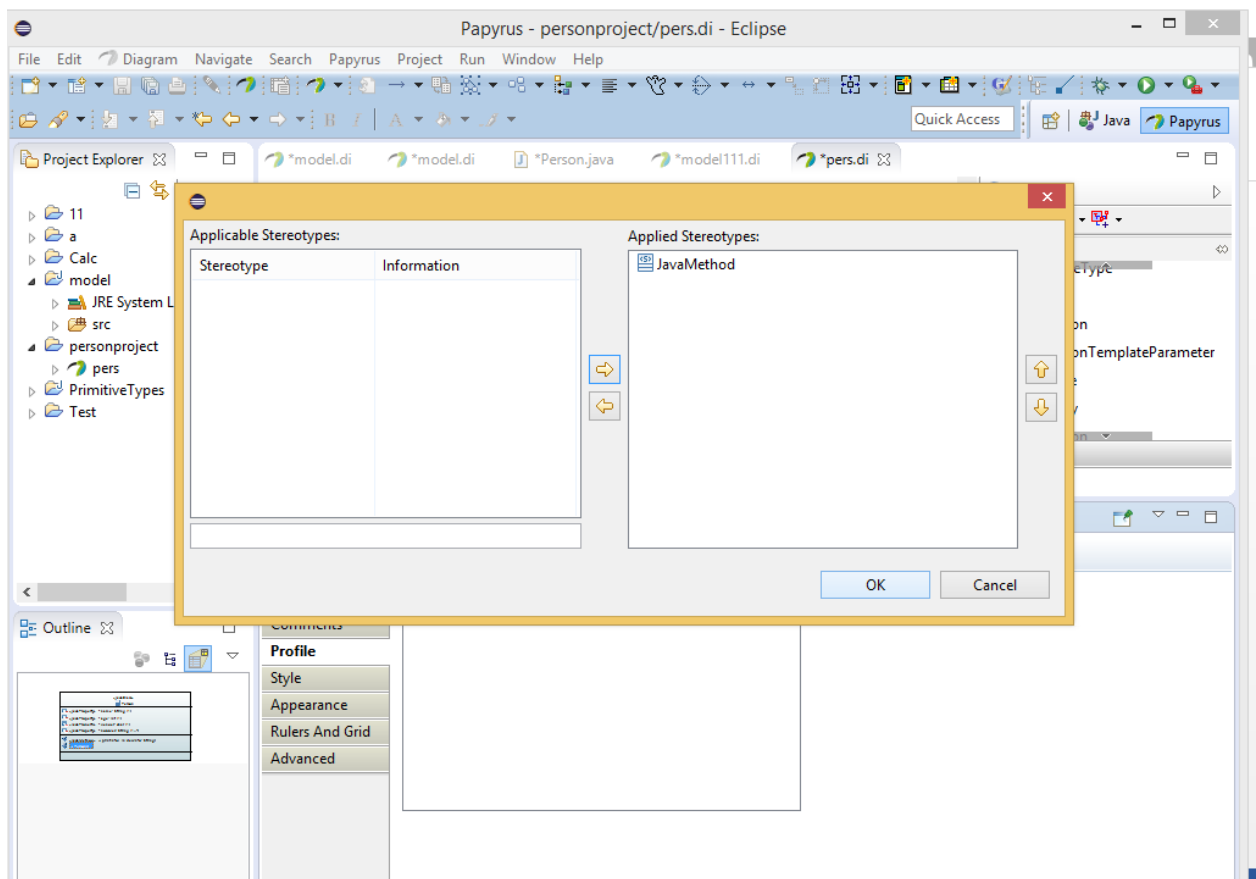
Покажемо на прикладі класу Person як описати конструктори за допомогою плагіна Rarurys. Отже, клас Person має наступний вигляд



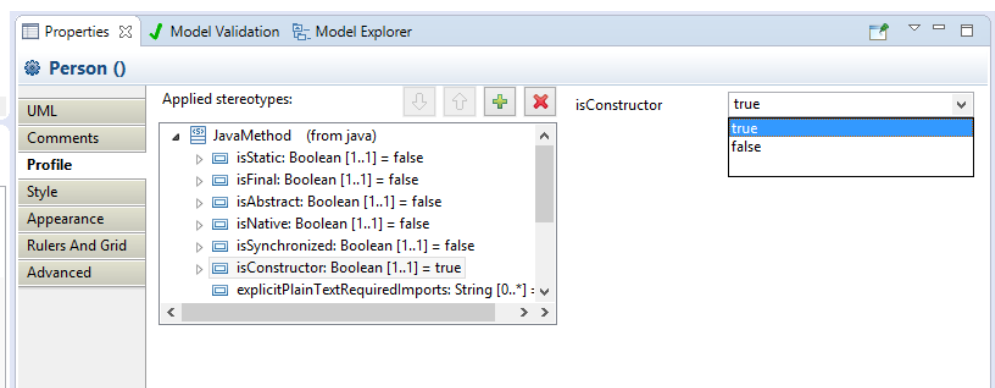
Додамо конструктор (нову операцію): і'мя Person- повинне співпадати з і'мям класу.



Далі необхідно перейти до закладки Profile ->Apply Stereotypes->Java Method.



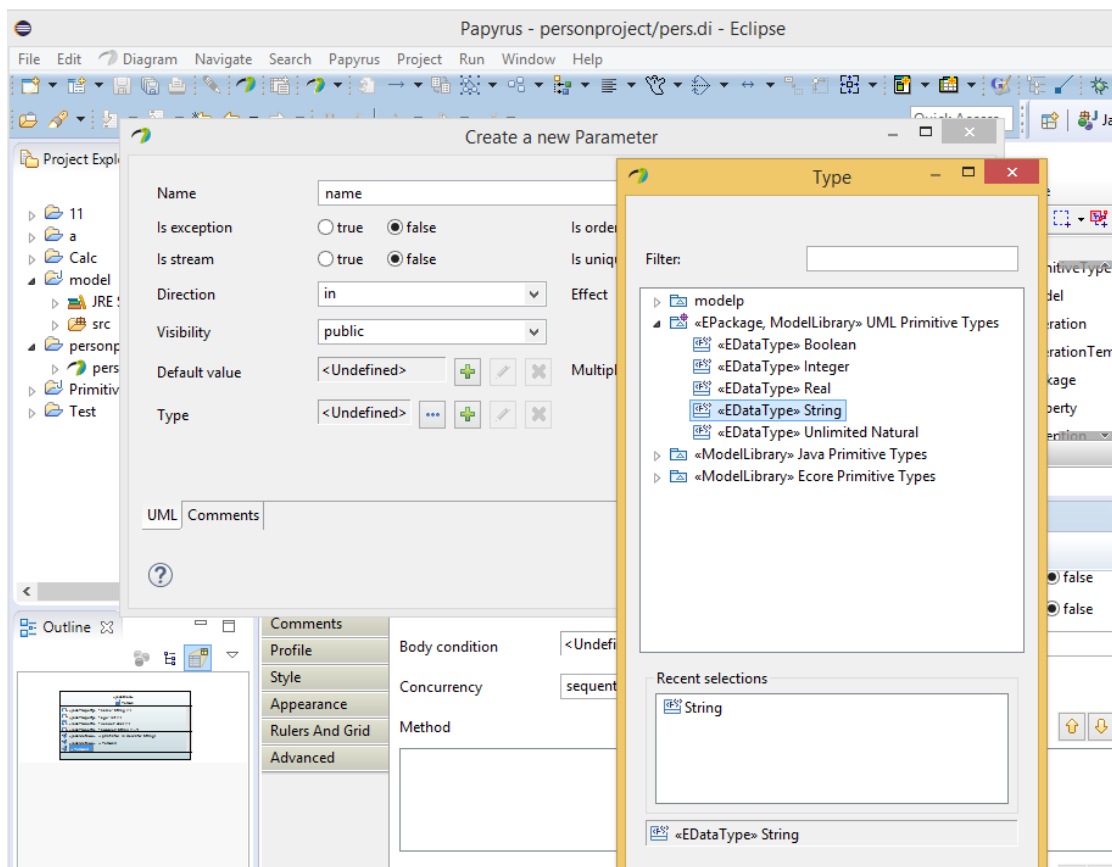
Після вибору стереотипу розгортаємо його повний список. Для isConstructor: Boolean [1..1]=false у правій частині закладки Profile обираємо ->true:



Таким чином ми можемо оголосити порожній конструктор. Додамо в наш клас другий конструктор, який буде мати кілька параметрів, а саме name, age та gender.

Переходимо на UML закладку: навиваємо конструктор ім'ям класу і додаємо такі параметри в Owned Parameter, як:

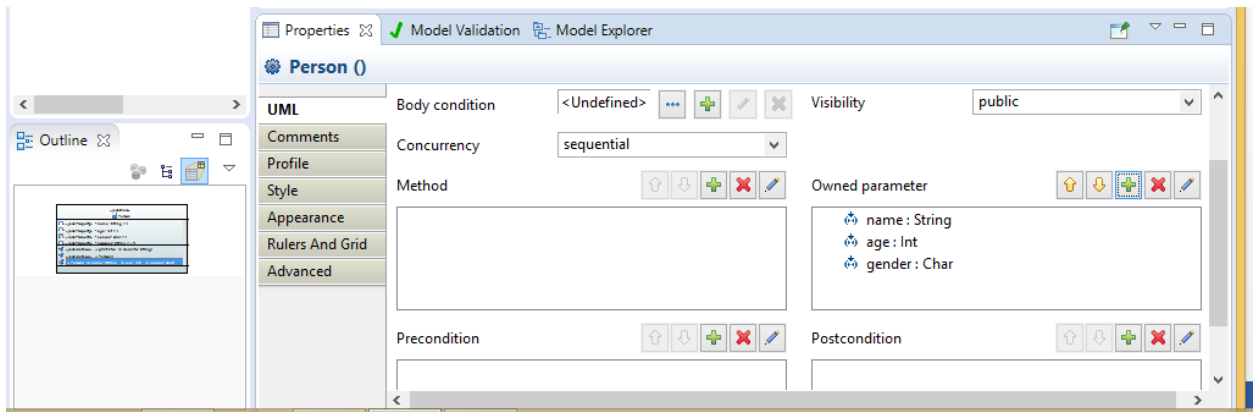
1 @param name



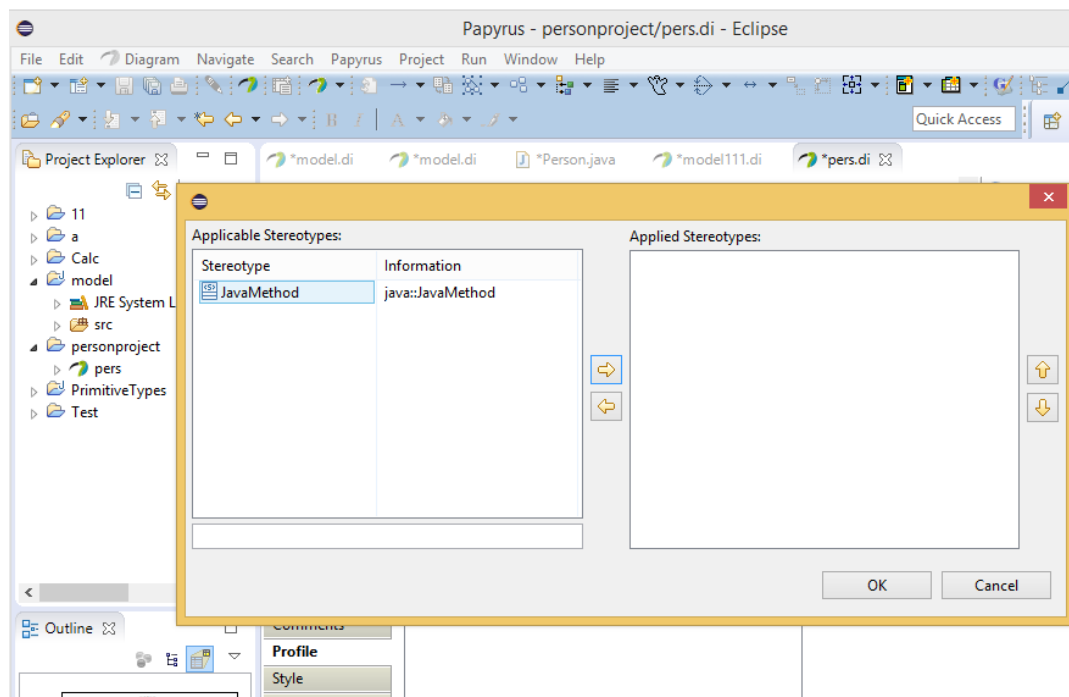
2 @param age : Int;

3 @param gender: String.

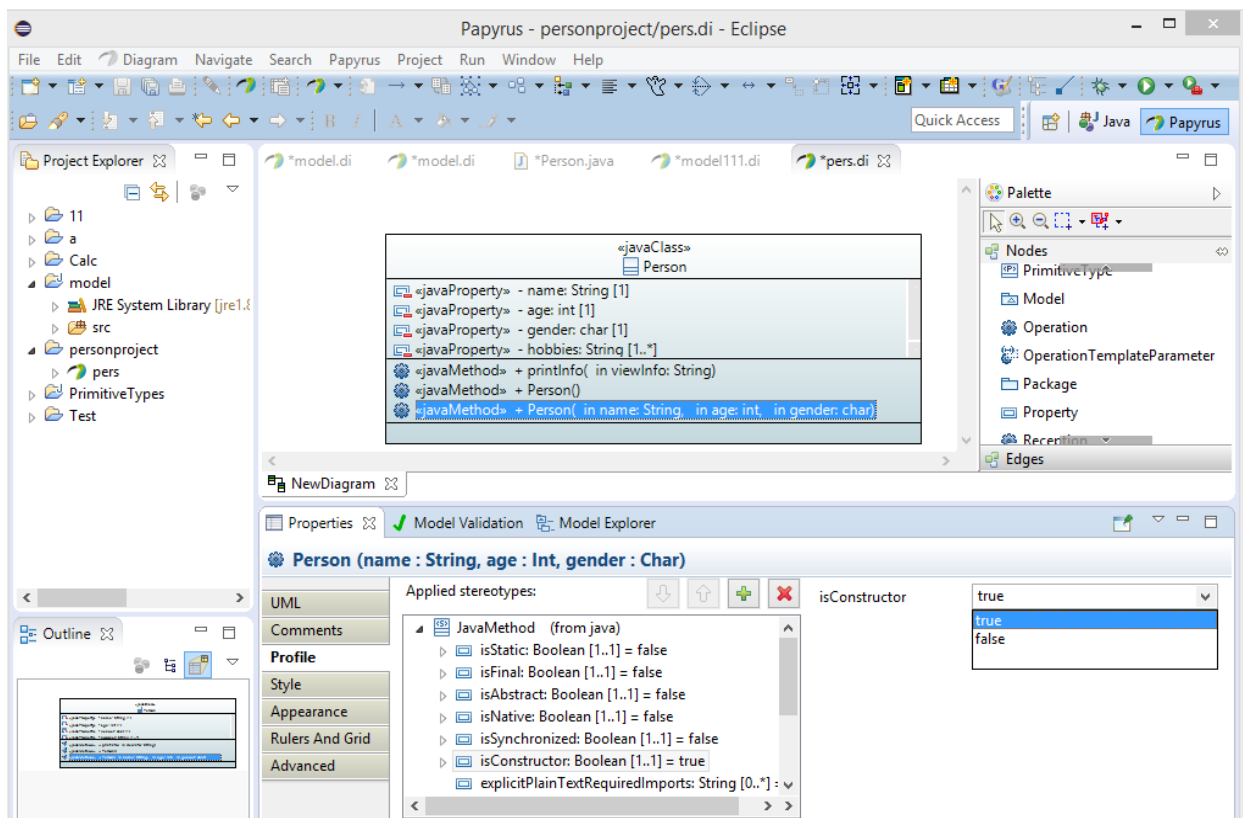
В результаті у вікні Owned Parameters з'являється список всіх оголошених параметрів:



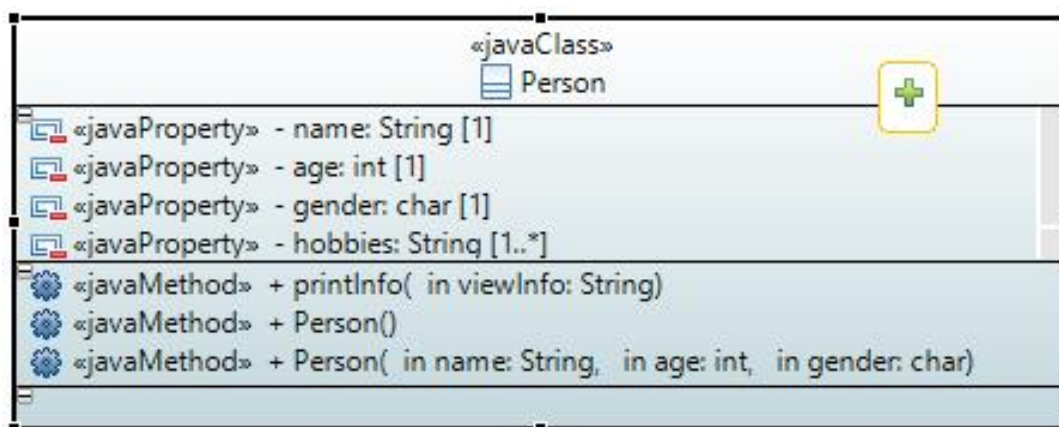
На останньому кроці для нашого конструктора з параметрами на закладці Profile-> Applied Stereotypes вибираємо JavaMethod



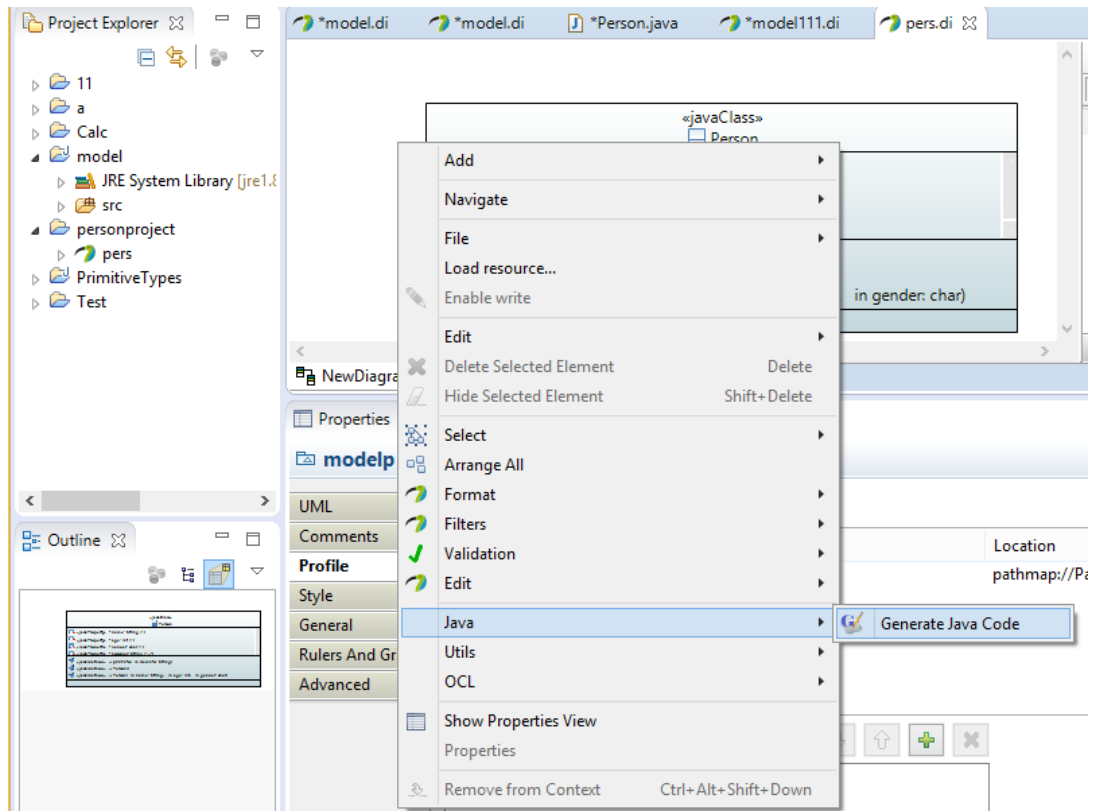
і в розгорнутому списку JavaMethods на закладці Profiles для isConstructor:
Boolean [1..1] = false міняємо на true



Таким чином UML діаграма класу Person приймає наступний вид:



Останнім шагом є генерація Java- коду для класу person: по полю моделі правим кліком обираємо Java->Generete Java Code



Лістинг коду:

```
import java.util.List;

public class Person {

    /**
     *
     */
    private List<String> hobbies;
    /**
     *
     */
    private String name;
    /**
     *
     */
    private int age;
    /**
     *
     */
    private char gender;
    /**
     * Getter of hobbies
     */
    public List<String> getHobbies() {
        return hobbies;
    }
}
```

```
/**
 * Setter of hobbies
 */
public void setHobbies(List<String> hobbies) {
    this.hobbies = hobbies;
}
/**
 * Getter of name
 */
public String getName() {
    return name;
}
/**
 * Setter of name
 */
public void setName(String name) {
    this.name = name;
}
/**
 * Getter of age
 */
public int getAge() {
    return age;
}
/**
 * Setter of age
 */
public void setAge(int age) {
    this.age = age;
}
/**
 * Getter of gender
 */
public char getGender() {
    return gender;
}
/**
 * Setter of gender
 */
public void setGender(char gender) {
    this.gender = gender;
}
/**
 *
 * @param viewInfo
 */
public void printInfo(String viewInfo) {
    // TODO Auto-generated method
}
/**
 *
 */
public Person() {
```

```

        // TODO Auto-generated method
    }
/**
 *
 * @param gender
 * @param name
 * @param age
 */
public Person(char gender, String name, int age) {
    // TODO Auto-generated method
}

```

Конструктори по замовчуванням мають наступний вигляд:

```

public Person() {
    // TODO Auto-generated method
}
/**
 *
 * @param gender
 * @param name
 * @param age
 */
public Person(char gender, String name, int age) {
    // TODO Auto-generated method
}

```

Внесемо кілька доповнень до кожного з них:

```

public Person() {
    this.name = null;
    this.age = 0;
    this.gender = 'U';
    this.hobbies = null;
}

public Person(char gender, String name, int age) {
    this.name = name;
    this.age = age;
    this.gender = gender;
    this.hobbies = null;
}

```

Постановка завдання і вихідні дані для варіантів:

Завдання:

1. Необхідно створити клас, що містить у своєму складі:
 - поля, в тому числі представлені у вигляді масиву,
 - відповідного варіанту типу даних;
 - кілька конструкторів, що дозволяють створювати об'єкти класу, поля яких заповнюються відповідно до типу і набору аргументів, переданих конструктору при ініціалізації об'єкта;
 - кілька реалізацій методу "порівняння", які перевантажують один одного і виконують різні дії в залежності від набору аргументів, переданих методом при його виклику;
 - поле, що зберігає текстовий опис об'єкта (коментар до нього).
2. Крім цього, у складі класу повинен бути метод "показати", що виводить значення полів об'єкта для контролю їхнього стану в наочному вигляді та інші методи, необхідність в яких може виникнути при вирішенні практичного завдання варіанта.
3. Вирішити запропоновані практичні завдання, розташували оператори в методі `main` створеного класу і оформивши їх у вигляді самостійного тестового класу. Порівняти отримані результати в тому і іншому випадку.
4. Для моделювання майбутнього класу використовувати плагін Eclipse-Parurys (for UML)

Варіант 1

Опишіть клас, що моделює клітку з папугами, представники якого зберігають інформацію про колір кожного птаха, їх кількість для даної клітки і величину споживаного на добу корму (за замовчуванням 20 грам на добу).

Передбачте можливість створення: "порожньої клітки" на задану кількість птахів; клітки шляхом прямого завдання кольору кожного птаха; клітки на задану кількість птахів з випадковим завданням кольору кожного птаха і клітки, що є копією деякої іншої, раніше створеної.

Представники класу повинні вміти порівнювати: себе на предмет визначення % заповнювання; себе і деяку іншу клітку на предмет рівності показника витрачання корму; себе і деякий колір на предмет пошуку номерів птахів відповідного кольору; себе і деяку масу на предмет встановлення факту "вистачить/ чи не вистачить" її для добового прохарчування птахів, що містяться в ній.

Допустимі кольори птахів: "червоний", "жовтий", "зелений", "блакитний", "нема птахи".

- Створіть клітку з 12-ти птахів, задавши колір кожної. Визначте, скільки корму з'їдять птахи за місяць і скільки вони з'їдять, якщо норма витрати корму збільшиться в два рази?
- Створіть клітку з 40 птахів з випадковим розподілом кольору.
- Визначте % заповненості клітки і кількість птахів кожного кольору. На скільки днів вистачить 3 кг корму для їх прохарчування?
- Створіть дві порожні клітки на 30-ть місць кожна і розсадіть в них птахів, комбінуючи "червоні з блакитними" і "жовті з зеленими". У скільки разів більше корму треба птахам 1-ої або 2-ої групи?

Варіант 2

Опишіть клас, що моделює бланк учасника лотереї, представники якого зберігають інформацію про загадані номери, максимально допустиме загадане число і загальну кількість загаданих номерів.

Передбачте можливість створення: "порожнього бланка" на задану кількість номерів; бланка, шляхом прямого завдання загаданого числа для кожної позиції; бланка, на задану кількість номерів з випадковим завданням цифр і бланка, що є копією деякого іншого, раніше створеного.

Представники класу повинні вміти порівнювати: себе на предмет визначення правильності заповнення (цифри не повинні перевищувати допустиму межу); себе і деякий інший бланк на предмет кількості збігів цифр по позиціях; себе і деяку цифру на предмет пошуку її позицій у бланку.

Примітка: у одному бланку може бути декілька рівних номерів.

- Створіть два бланки, що містять 10-ть номерів у діапазоні 1...5. Перевірте правильність їх заповнення. Створіть два порожніх бланка і скопіюйте в 1-й числа "1 і 3", а в 2-й числа "2 і 4" в порядку їх слідування в базовому бланку.
- Задайте третій бланк на 10-ть номерів, в якому цифри задані випадково в діапазоні 50...200. Скопіюйте його в 4-й бланк, встановіть для нього параметр "максимальної цифри" рівної 5-ть і визначайте номери, що виходять за допустиму межу і зменшуйте їх в 2 рази до тих пір, поки бланк не буде відповідати нормі.
- Порівняйте отриманий бланк з двома, заданими в першому пункті на предмет збігу номерів по позиціях.

Варіант 3

Опишіть клас, що моделює грядку з капустою, представники якого зберігають інформацію про масу кожної посадженої рослини, максимально

допустимому їх кількість для даної грядки і показник, що характеризує швидкість набору маси (за замовчуванням 10 грам на добу).

Передбачте можливість створення: "порожньої грядки" на задану кількість рослин; грядки шляхом прямого завдання параметра кожної рослини; грядки на задану кількість рослин з випадковим завданням маси кожної в зазначеному діапазоні і грядки, що є копією деякої іншої, раніше створеної з урахуванням збільшення маси рослин за вказаний час.

Представники класу повинні вміти порівнювати: себе і деяку іншу грядку, на предмет рівності показника "сумарна маса рослин/їх загальна кількість"; себе і деяку масу, на предмет пошуку номерів рослин більшої маси; себе і деякий діапазон мас, на предмет пошуку кількості рослин, що потрапляють до нього.

- Створіть 1-у грядку з капустою, задавши масу кожної з 15 рослин в інтервалі 20...50 грам кожна. З інтервалом у два дні перевіряйте ситуацію на грядці і видаляйте рослини з вагою більше 200 грам (замініть масу на нуль). Скільки днів знадобитися для повного дозрівання всієї грядки при швидкості набору ваги 10 грам на добу?
- Створіть 2-у грядку випадкового виду на 14 рослин масою 30...80 грам і простежте тенденцію зростання за місяць з інтервалом 5 днів при швидкості набору ваги 5 грам на добу. Скільки рослин з масою 150...200 грам будуть рости на грядці в кожен контрольний день?
- Створіть 3-ю порожню грядку на 10 рослин. Посадіть в 1-у позицію капусту масою 5 грам. У день коли її маса подвоїться - садіть в наступну позицію капусту з масою, в половину меншою і щодо неї повторюйте даний алгоритм до повного засадження грядки. Швидкості набору ваги прийняти 15 гр / д. У скільки разів буде відрізняться маса капусти в 1-й і 10-й позиції?
- Порівняйте між собою 2-у і 3-ю грядку в їх кінцевому вигляді?

Варіант 4

Опишіть клас, що моделює план руху людини по місцевості, представники якого зберігають інформацію про напрям кожного її кроку, а також максимальне число кроків для даного маршруту.

Передбачте можливість створення: "порожнього маршруту" на задану кількість кроків; маршруту, що містить перелічені назви кроків; маршруту на задану кількість кроків, напрям яких задано випадковим чином; маршруту який є точною копією іншого, раніше створеного.

Представники класу повинні вміти порівнювати: себе самого, на предмет визначення чи буде його маршрут рухом по замкнутому колу; себе і деякий напрям, на предмет пошуку номерів кроків у маршруті, зроблених у цьому напрямку; себе і початкову координату, на предмет обчислення нової координати, в якій опиниться людина пройшовши її маршрут із заданої точки; себе і деякий маршрут, на предмет визначення в якому з них більше "порожніх" кроків.

Кроки можуть бути описані: "на місці", "північ", "південь", "захід", "схід".

- Задайте маршрут руху людини з 15 кроків і встановіть її місце розташування при базовій точці виходу (5, -2);
- Створіть точно такий же маршрут, замініть "північ" на "на місці", а "на місці"- на "схід" і встановіть її місце розташування при тій же базовій точці виходу, а також порівняйте маршрути, на предмет інтенсивності руху кожного.
- Задайте випадковий маршрут з 50 кроків і визначте, чи буде він "рухом по колу". Створіть 2 порожніх маршрути на 50 кроків і скопіюйте в 1-й всі кроки на "північ і схід", а в 2-й відповідно "південь і захід" у черговості їх слідування в базовому маршруті. Визначте положення людини, яка вийшла із точки (0,0) по 1-му маршруту, а після його завершення пройшла по 2-му.

Варіант 5

Опишіть клас, що моделює стійку для компакт дисків з фільмами, представники якого зберігають інформацію про жанр кожного диска, а також максимальне число дисків для даної стійки.

Передбачте можливість створення: "порожньої стійки" на задану кількість дисків; стійки, яка містить зазначені диски; стійки на задану кількість дисків, клітинки якої заповнені дисками, жанр яких заданий випадковим чином; стійки, що є точною копією іншої, раніше створеної.

Представники класу повинні вміти порівнювати: себе самого, на предмет визначення кількості порожніх клітинок; себе і деякий жанр, на предмет пошуку номерів клітинок, які містять диски зазначеного жанру; себе, жанр і деяке число, на предмет встановлення факту наявності фільмів даного жанру кількістю більшою, ніж задана; себе і деяку іншу стійку, на предмет рівності показника "максимальний об'єм стійки/кількість заповнених клітинок".

Асортимент жанрів приблизно наступний: "комедія", "жахи", "мелодрама", "бойовик", "фантастика".

- Створіть стійку, яка містить фільми зазначених жанрів (безпосередньо вкажіть 10-15 фільмів або порожніх клітинок " " в довільному порядку і кількості) вказаного об'єму. Підрахуйте кількість фільмів кожного жанру та їх відсотковий склад відносно загального числа фільмів в стійці?
- Створіть ще одну таку саму стійку і видаліть з неї "бойовики" і "жахи". Скільки порожніх клітинок тепер є на стійці?
- Створіть стійку на 30 компактів з фільмами випадкового жанру і розкладіть диски в 5 порожніх стійок (на 5 дисків кожна) за жанрами.

Скільки фільмів якого жанру залишиться у вихідній стійці?

Варіант 6

Опишіть клас, що моделює довідку про доходи за деякий період часу, представники якого зберігають інформацію про суму виплат в кожен місяць, а також кількість місяців охоплених даним документом.

Передбачте можливість створення: "порожньої довідки" на задану кількість місяців; довідки, яка містить конкретні значення доходів по кожному місяцю; довідки на заданий період часу, де значення виплат задані випадковим чином у вказаному діапазоні; довідки, яка є точною копією іншої, раніше створеної.

Представники класу повинні вміти порівнювати: себе самого, на предмет визначення середньої величини виплати за даний період; себе і деякий чисельний діапазон, на предмет пошуку номерів місяців для яких виплата потрапляє в межі вказаного діапазону; себе і деяку іншу довідку, на предмет з'ясування факту чи показник середнього значення для даної довідки "більше/менше" по відношенню до іншої.

- Створіть довідку № 1 зарплати за місяць співробітника деякої фірми у відповідних межах і аналогічну довідку № 2. Визначте до кого з вас перейде працювати ця людина в наступному році і наскільки вагомими будуть його аргументи?
- Створіть довідку № 3 за 2 роки, де виплати задані випадково в діапазоні 200...500 грн і визначте середнє значення за даний період і в які місяця сума знаходилася в діапазоні "середнє (+/-) 10% "?
- Створіть на основі довідки № 3 довідку № 4, де замініть нулями виплати менші 250 і більші 450 і визначте наскільки змінився середній показник?

Варіант 7

Опишіть клас, що моделює алгоритм роботи автомата, який переміщує вантажі по деякому маршруту, представники якого зберігають інформацію про

характер кожної операції, а також їх максимальну кількість для даного алгоритму і початкову координату положення автомата.

Предбачте можливість створення: "порожнього алгоритму" на задану кількість операцій; алгоритму, що містить перелік зазначених операцій; алгоритму на задану кількість операцій, характер яких заданий випадковим чином; алгоритму, який є точною копією іншого, раніше створеного.

Представники класу повинні вміти порівнювати: себе самого, на предмет визначення чи є його алгоритм циклічно замкнутим; себе і деяку операцію, на предмет пошуку номерів подібних операцій в алгоритмі; себе і деяку координату, на предмет визначення факту збігу її з маршрутом руху; себе і деякий маршрут, на предмет визначення в якому з них більше "порожніх" кроків.

Кроки можуть бути описані: "на місці", "вгору", "вниз", "вперед", "назад".

- Створіть алгоритм роботи автомата, який повинен перенести вантаж з точки, що знаходиться на висоті 4 пункти в трьох кроках позаду точки вихідного положення в точку, що знаходиться на висоті 3 пункти в 4-х кроках від базової точки з поверненням в початковий стан. Перевірте даний алгоритм на циклічність.
- Створіть випадковий алгоритм на 20 кроків і перевірте крайні положення його зміщення відносно базової точки. Чи буде перебувати точка, розташована на висоті 3 пункти у 2 кроках попереду базової точки знаходиться на шляху руху автомата?
- Створіть алгоритм - копію випадкового і два порожніх такого ж розміру. Перенесіть в них операції з вихідного за принципом: у 1-й - операції у вертикальному напрямку, а в 2-й - у горизонтальному. У якому з отриманих алгоритмів буде більше "порожніх" операцій?

Варіант 8

Опишіть клас, що моделює сумку з інструментом, представники якого зберігають інформацію про найменування кожного інструменту, а також їх максимальне число для даної сумки.

Передбачте можливість створення: "порожньої сумки" на задану кількість інструментів; сумки, яка містить перелічені назви інструментів; сумки на задану кількість інструментів, асортимент яких заданий випадковим чином; сумки, що є точною копією іншої, раніше створеної.

Представники класу повинні вміти порівнювати: себе самого, на предмет визначення % заповненості; себе і деякий інструмент, на предмет пошуку кількості екземплярів даного виду; себе і два найменування інструменту, на предмет встановлення факту "більше/менше" інструментів 1-го виду, ніж 2-го; себе і деяку іншу сумку, на предмет рівності показника "максимальний об'єм сумки/кількість покладених інструментів".

У сумці можуть перебувати: "викрутка", "кусачки", "молоток", "гайковий ключ", "ніж".

- Створіть сумку на 20 предметів з випадковим характером їх наповнення і іншу, точно таку ж, в якій знайдіть і видаліть "кусачки" і порівняйте показник заповненості сумок.
- Задайте сумку шляхом конкретної вказівки інструменту, що знаходиться в ній і порожніх клітинок і визначте інструмент якого виду в ній переважає і інструмент, якого менше.
- Створіть випадкову сумку на 50 предметів і розсортуйте за двома порожніми сумкам на 20 предметів "викрутки і ножі" і "молотки і гайкові ключі", і визначте наскільки в першій сумці предметів більше, ніж у другій.

Варіант 9

Опишіть клас, що моделює товарний потяг, представники якого, зберігають інформацію про назву вантажу і кількість вагонів в потязі.

Передбачте можливість створення: “порожнього потяга” на задану кількість вагонів; потяга, що перевозить перелічені назви вантажів; потяг на задану кількість вагонів, вміст яких заданий випадковим чином; потяга, що є точною копією, створеного раніше.

Представники класу повинні вміти порівнювати: себе самого на предмет % наповненості; себе і вагони з деяким вантажем на предмет пошуку кількості вагонів даного вантажу; себе і вагони з двома назвами вантажу, на предмет встановлення факту “більше / менше” вагонів одного вантажу, ніж іншого; себе і деякий інший потяг на предмет рівності показника “максимальний об’єм потяга/кількість причіпних вагонів”.

Товарний потяг може складатися із вагонів з таким вантажем: “бензин”, “банани”, “вугілля”, “зерно”, “добрива”.

- Створіть товарний потяг на 22 вагони з випадковим наповненням вантажем та інший, точно такий же, в якому знайдіть та видаліть вагони з “зерном” і порівняйте показники наповненості потягів.
- Задайте потяг шляхом конкретного завдання вантажу, що він перевозить і пустих вагонів і визначте вагони з яким вантажем в ньому переважають і яких вагонів менше.
- Створіть випадковий потяг на 37 вагонів і розсортуйте по двох порожніх потягах на 15 вагонів “бензин та вугілля” і “банани та добрива” і визначте на скільки в першому потязі вагонів більше ніж в другому.

Варіант 10

Опишіть клас, що моделює розрахункову книжку по сплаті за комунальні послуги, представники якого зберігають інформацію про суму боргу за місяць і кількість місяців в книжці.

Передбачте можливість створення: “пустої книжки” на задану кількість місяців; книжки, де зазначені конкретні значення суми боргу по кожному місяцю; книжки на заданий період часу, де значення сум боргу задані випадковим чином; книжки, що є точною копією створеної раніше.

Представники класу повинні вміти порівнювати: себе самого на предмет визначення середньої величини суми боргу в заданий період; себе і деякий числовий діапазон на предмет пошуку номерів місяців для яких сума боргу попадає в границі вказаного діапазону; себе і іншу книжку на предмет встановлення факту чи показник середнього значення суми боргу для даної книжки “більше / менше” по відношенню до іншої.

- Створіть розрахункову книжку №1 де суми боргу за місяць будуть в певному діапазоні і аналогічну книжку №2. Визначте де суми боргу будуть більшими.
- Створіть розрахункову книжку №3 за 2012 рік, де суми боргу задані випадково в діапазоні 35...60 грн и визначте середнє значення за даний період і в які місяці сума боргу знаходилася в проміжку “середнє значення (+/-) 5%”.
- Створіть розрахункову книжку №4 на основі книжки №3, де замініть нулями суму боргу менше 20 грн і більше 50 грн і визначте, на скільки зміниться середній показник.

Контрольні питання

1. Дайте визначення конструктора.
2. Які особливості оголошення конструктора?
3. Для чого потрібні конструктори?
4. Що таке перевантаження методів?
5. Як передати масив в якості аргумента метода?

ЛАБОРАТОРНА РОБОТА №3

«Створення UML – діаграми для Спадкування членів класу.
Перевизначення методів».

Мета: Ознайомитися з принципами побудови ієрархії спадкування.
Навчитися будувати підкласи, звертатися до методів суперкласу і перевизначати методи. Вміти створювати і працювати з представниками всіх рівнів ієрархії.

Рекомендації до виконання роботи (частина 1):

Принципи побудови ієрархії класів

У випадках, коли створюється декілька класів між якими простежується логічний зв'язок "від загального до приватного", має сенс створити ієрархічну структуру на чолі якої поставити клас, що володіє загальним набором властивостей і методів для решти класів - членів ієрархії,

які успадкують їх і розширять склад класу шляхом додавання до нього додаткових елементів для вирішення більш приватних завдань.

Прикладом сказаного може служити, наприклад, структура "точка - точка з параметром - (маса, заряд, колір та ін) - точка з випадковими полями". Очевидно, що метод зі складу базового класу "точка", наприклад для визначення відстані між даною точкою і деякої іншої, буде точно таким же, незалежно від того чи володіє точка якою-небудь властивістю, додатково характеризує її стан і від того яким способом були задані її координати:

```
double dist (point p) {
return Math.sqrt (p.x*p.x+p.y*p.y) ;
}
```

У цьому випадку класи-спадкоємці успадкують цей метод, так само як і поля, що зберігають інформацію про положення точки на площині (x, y) і немає необхідності кожного разу повторювати їх опис - досить вказати ключове слово "extends" при описі класу-спадкоємця, вказати наслідуваний клас і всі нестатичні компоненти базового класу будуть успадковані, наприклад :

```
class paramPoint extends point {}
```

У свою чергу в складі класу paramPoint з'являється додаткове поле (massa, charge, color...) і методи для роботи з цим полем (порівняння мас даної точки і деякої іншої, встановлення фактора притягання/відштовхування зарядів або конвертація кольору по RGB складовим...). Водночас зберігається можливість викликати для цих точок методи роботи з координатами так, як якщо б вони були явно описані в класі paramPoint, наприклад :

```
paramPoint pp1 = new paramPoint (10, -10,2);
paramPoint PP2 = new paramPoint (20,0,4);
System.out.println ("Distanse:" +pp1.dist(pp2));
```


Звернення до методів суперкласу

Конструктори підкласів можуть викликати методи суперкласу використовуючи ключове слово "super", наприклад:

```
paramPoint (point p, int massa){
    super(px, py);
    this.massa = massa;
}
```

в рядку super (px, py) відбувається виклик конструктора базового класу, який ініціалізує поля об'єкту за переданими значеннями координат (x, y). Слід зауважити, що в тілі конструктора подібний виклик повинен йти самим першим рядком, інакше компілятор видає помилку. За допомогою super можна, у разі необхідності, викликати не тільки конструктори, а й інші методи базового класу, наприклад :

```
double calculation (paramPoint pi, paramPoint p2) {
    return (super.dist (pi) + super.dist (p2))/2;
}
```

двічі викликає метод dist() суперкласу, для визначення середньої відстані між даною точкою і двома іншими.

Перевизначення методів

У тих випадках, коли однойменний метод підкласу реалізується інакше ніж це було задумано в суперкласі, його можна перевизначити, записавши його нову реалізацію.

Наприклад, метод toString викликається за замовчуванням системою при спробі вивести на друк посилання на об'єкт командою System.out.println(), знаходиться в класі Object, який стоїть на чолі всієї ієрархії класів і за

замовчуванням успадковується усіма існуючими класами, може бути перевизначений для прикладу з точками наступним чином:

```
public String () {
    System.out.println ("This point (" + x + ", " + y + ")");
}
point p1 = new point (3, -5);
System.out.println ("First point " + p1);
```

Тепер можна отримати інформацію про створений об'єкт у вигляді First point(3,-5). Модифікатор доступу "public" обов'язковий тому, що перевизначені методи не можуть звужувати можливість доступу до методу, а оскільки в класі Object метод toString() описаний як public, то і в класахспадкоємцях він повинен залишатися таким.

Статичні методи і об'єкти

Статичні методи можуть приймати в якості аргументів як стандартні типи даних, так і посилання на об'єкти. Також вони можуть повертати об'єкти, які потім можна присвоювати об'єктам такого ж типу. Наприклад:

```
point static halfpoint (point p1, point p2) {
    point half = new point ();
    half.x = ( p1.x + p2.x ) / 2;
    half.y = ( p1.y + p2.y ) / 2; return half ;
}
```

Статичний метод halfpoint отримує посилання на дві точки, створює нову точку з нульовими координатами, присвоює їм значення середини інтервалу по кожній осі і повертає отриману точку.

```
point start = new point (2, 0);
```

```

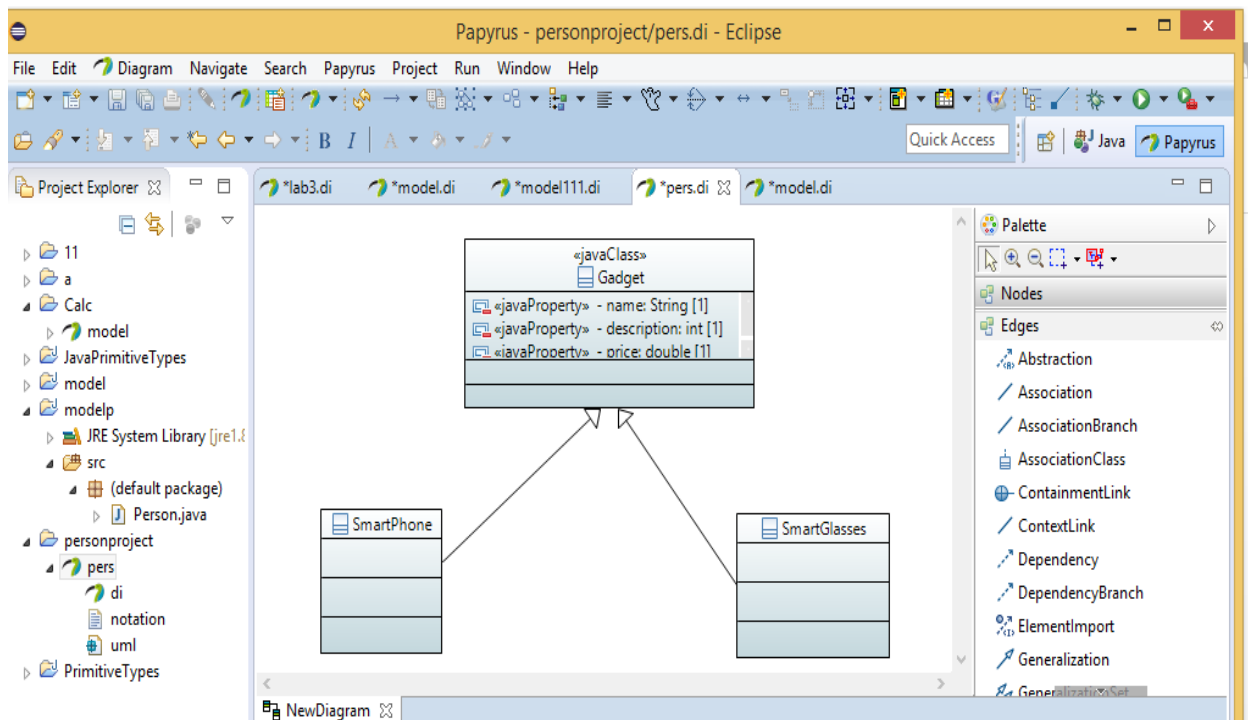
point end = new point (10,-8);
point phalf;
phalf = halfpoint (start, end);
System.out.println ("First point " + p1);

```

Дані операції створять точку (6,-4), що знаходиться в середині заданого відрізка.

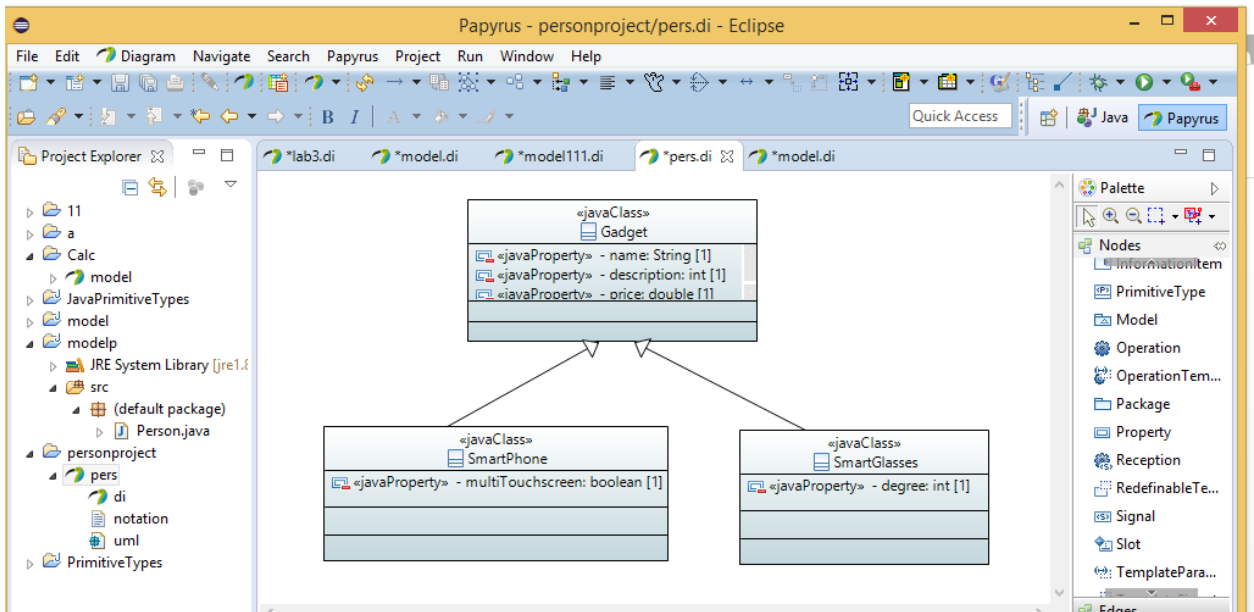
Рекомендації до виконання роботи (частина 2): моделювання за допомогою плагіну Papyrus.

Для прикладу генерації коду з відношенням успадкування між класами побудуємо в плагіні Papyrus наступну діаграму класів:



де Gadget - клас-батько, а класи SmartPhone і SmartGlasses його нащадки.

Крім успадкованих атрибутів від класу-батька у кожного нащадка будуть свої атрибути: так у нащадка - SmartPhone є унікальний атрибут multiTouchscreen: Boolean, а для нащадка SmartGlasses атрибут degree: int. Діаграма класів для генерації коду має такий вигляд (як описувати атрибути і операції див. першу лабораторну роботу!)



Після запуску згенерувати код в закладці Project Explorer з'явиться список згенерованих класів.

Код для класу Gadget:

```

public class Gadget {

    /**
     *
     */
    private String name;
    /**
     *
     */
    private double price;
    /**
     *
     */
    private int description;
    /**
     * Getter of name
     */
}
  
```

```

public String getName() {
    return name;
}
/**
 * Setter of name
 */
public void setName(String name) {
    this.name = name;
}
/**
 * Getter of price
 */
public double getPrice() {
    return price;
}
/**
 * Setter of price
 */
public void setPrice(double price) {
    this.price = price;
}
/**
 * Getter of description
 */
public int getDescription() {
    return description;
}
/**
 * Setter of description
 */
public void setDescription(int description) {
    this.description = description;
}
}
}

```

Код для класу SmartPhone:

```

public class SmartPhone extends Gadget {

    /**
     *
     */
    private boolean multiTouchscreen;

    /**
     * Getter of multiTouchscreen
     */
    public boolean getMultiTouchscreen() {
        return multiTouchscreen;
    }
}

```

```

/**
 * Setter of multiTouchscreen
 */
public void setMultiTouchscreen(boolean multiTouchscreen) {
    this.multiTouchscreen = multiTouchscreen;
}
}

```

Код для класу SmartGlasses:

```

public class SmartGlasses extends Gadget {

    /**
     *
     */
    private int degree;

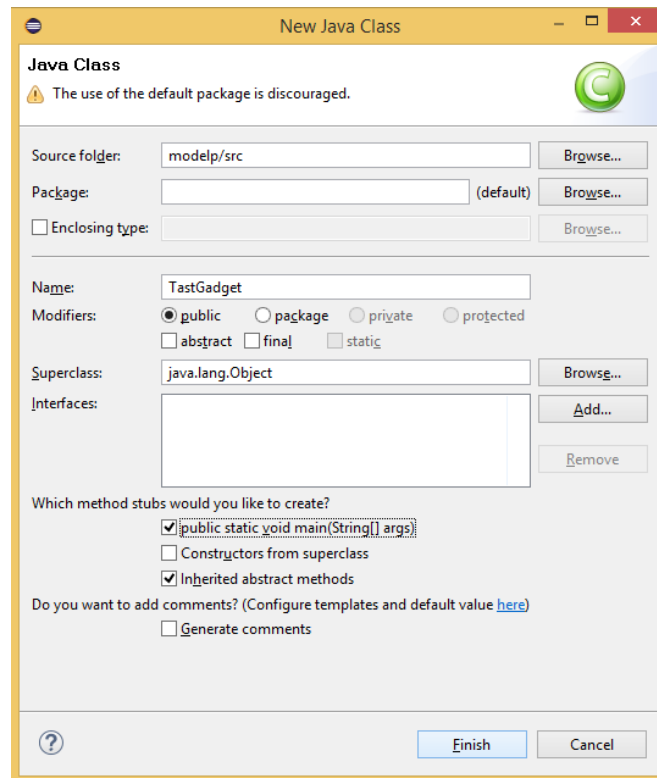
    /**
     * Getter of degree
     */
    public int getDegree() {
        return degree;
    }

    /**
     * Setter of degree
     */
    public void setDegree(int degree) {
        this.degree = degree;
    }

}

```

Для демонстрації роботи відношення «успадкування» створемо ще один клас TestGadget



```
public class TastGadget {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

внесем в его код несколько дополнений:

```
public class TastGadget {
    public static void main(String[] args) {
        SmartPhone phone1 = new SmartPhone();
        phone1.setName("Samsung Galaxy S3");
    }
}
```

Постановка завдання і вихідні дані для варіантів:

Завдання:

1. Описати базовий клас визначивши, яким набором полів повинні володіти його представники для достовірної побудови моделі і описати зазначені у варіантах методи. Передбачити кілька конструкторів для створення представників класу.
2. Описати підклас 1, що спадкує базовий клас. Розширити його набір полів, відповідно до методів, що додаються до його складу і описати ці методи з використанням операцій виклику методів суперкласу. Додати необхідні конструктори, що використовують у разі потреби звернення до конструкторів суперкласу. Перевизначити методи базового класу при необхідності.
3. Описати підклас 2, який спадкує підклас 1 і відрізняється від нього тільки тим, що ініціалізує деякі поля об'єкту випадковими величинами в зазначеному при виклику конструктора діапазоні.
4. Перевизначити метод `toString` для кожного класу.
5. Описати тестовий клас, у методі `main` якого, реалізувати запропоновані практичні додатки, що використовують створені раніше класи. Багаторазово використані операції оформити у вигляді окремого статичного методу, якому передаються по необхідні посилання на об'єкти і чисельні параметри і який сам повертає посилання на об'єкт.
6. Для моделювання майбутньої діаграми класів використовувати плагін Eclipse-Parurys (for UML)

Варіант 1

Базовий клас і його методи: коло на площині; обчислення площі і довжини кола.

Підклас 1 і його методи: об'ємний диск; обчислення об'єму та площі бічної поверхні.

Підклас 2: об'ємний диск з випадково заданим розміром .

Практичні додатки:

Визначте яка повинна бути кругла листова заготовка для того, щоб з неї можна було виготовити 50 однакових круглих деталей. Коефіцієнт щільності розміщення прийняти рівним 0,75. Аналогічні дії повторіть для 100, 125, 150 і 200 деталей.

Перевірте, чи зможуть розміститися на аркуші метрового радіусу 10-ть дисків випадкового розміру з радіусом в діапазоні 5...10 см? Скільки таких дисків необхідно для максимального заповнення площі аркуша?

Диск з відношенням радіуса до товщини, як 1:5 пофарбували синьою фарбою з боків і білою обидві основи. Аналогічно вчинили з диском 5:1. У якому випадку, якої фарби і в скільки разів більше знадобилося?

Варіант 2

Базовий клас і його методи: процес нагрівання деякого тіла масою m з речовини з коефіцієнтом теплового нагрівання $coef$ на температуру $difT$ вимагає витратити енергії $Q=m*coef*difT$. Модель даного процесу повинна вміти обчислювати Q по $difT$ і навпаки, а також порівнювати об'єкти між собою на предмет відмінності величин Q при заданому $difT$.

Підклас 1 і його методи: відрізняється можливістю моделювати процес шляхом завдання не різниці температури, а її абсолютних величин у початковій t_1 і кінцевій t_2 точці, а також враховувати ККД нагріваючого пристрою $kpd=100*Q_{real}/Q$ в%.

Підклас 2: при створенні об'єкта на основі переданого, змінює бажаний ККД на величину деякого відхилення в діапазоні (+ / -) %.

Практичні додатки:

На скільки градусів нагріються бруски масою 35 кг з матеріалів з $coef=13,26; 34,62$ і $5,15$?

Яка кількість тепла буде потрібна для нагріву матеріалу з $coef=34,45$ масою 10, 20 і 50 кг від 35 до 99°C при ККД печі 80%.

У трьох реальних печах з ККД 75% і початковою температурою 20°C нагріваються заготовки з матеріалу з $\text{coef}=10,23(+/-)5\%$ і масою 30 кг. До якої температури кожна піч змогла нагріти заготовки, витративши на це рівну кількість тепла?

Варіант 3

Базовий клас і його методи: стрілочний струмовий індикатор. Характеризується кількістю поділок шкали N_{max} і струмом повного відхилення стрілки I_{max} (відповідає максимальній відмітці шкали). Може по проміжному току I_{real} видавати відповідне йому показання шкали N_{real} , а також зворотна дія - за показанням шкали видавати струм, що викликав відповідне відхилення стрілки.

Підклас 1 і його методи: стрілочний струмовий індикатор з вхідним дільником струму. Додається коефіцієнт $k=I_{\text{input}}/I_{\text{real}}$, метод для перерахунку I_{real} з урахуванням дільника і відповідним чином уточнюються методи базового класу.

Підклас 2: реальний індикатор з дільником. Залежно від допустимого для нього % похибки змінює струм повного відхилення ідеального приладу (на основі якого він створюється) на $(+ / -) \%$ від I_{max} .

Практичні додатки:

Перевірте правильність роботи одно діапазонного стрілочного індикатора зі шкалою 50 ділень і струмом повного відхилення 100 мА, шляхом перебору за 100 кроків всього допустимого діапазону вхідного струму.

Індикатор зі шкалою 500 поділок і струмом повного відхилення 50 мА вимірює струм 90 мА при установці вхідних дільників на значення 1, 2 і 5. Які показники можна прочитати на шкалі в кожному випадку ?

Виготовлено три однотипних індикатора зі шкалою 50 ділень на струм шкали 50 мА $(+/-)5\%$. Наскільки відрізнятяться їх свідчення при вимірі струму рівному 10 мА?

Варіант 4

Базовий клас і його методи: точка, що рухається прямолінійно з постійною швидкістю v з координатою X_0 в початковий момент часу. Визначення пройденої за час t відстані, обчислення координати нового положення точки через час t , а також визначення часу необхідного для проходження заданої відстані.

Підклас 1 і його методи: точка, що рухається з прискоренням A . Пройдений шлях враховує додаткову складову $a \cdot t^2 / 2$.

Підклас 2: на основі переданої точки створює таку, яка відрізняється від неї тим, що початкова швидкість змінюється на величину зазначеного відхилення в діапазоні $(\pm)\%$.

Практичні додатки:

Два автомобілі почали рух з точок з координатами 0 і 5 км від початку відліку зі швидкостями 100 і 80 км/год в одному напрямку. Простежуючи зміну їх координат з інтервалом в 5 хв визначте координату точки обгону і відповідний їй момент часу.

Рухаючись в парі зі швидкістю 30 км/год два велосипедиста почали прискорюватися на спуску, причому 1-й з прискоренням 10 км/год², а 2-й - 8 км/год². Визначте розрив між ними через 1 хвилину.

По мішені, що знаходиться на відстані 10 метрів було випущено дві реальні стріли. Нормована швидкість вильоту стріли 100 км/год $(\pm) 15\%$. Визначте, який стрілі знадобилося менше часу для досягнення цілі.

Варіант 5

Базовий клас і його методи: моделювання витрат фарби для друку на аркуші паперу розміром (x, y) тексту залежно від кількості символів n , їх розміру (a, b) і норми витрати фарби на один см² друкованого тексту, визначення можливості "так/ні" розміщення заданого об'єму тексту на аркуші.

Підклас 1 і його методи: моделювання друку багатосторінкового документу з урахуванням коефіцієнту втрати фарби на валиках принтеру.

Підклас 2: моделювання реального принтера з відхиленням коефіцієнта втрати фарби щодо нормативу на величину в діапазоні зазначеного відхилення (+ / -) %.

Практичні додатки:

Визначте витрати фарби для друку тексту зі 150 символів розміром 3x5мм, при нормі витрати фарби 0,1 г/см² на аркуші паперу формату А4. Чи поміститься на даному аркуші в 2 рази більша кількість символів.

Скільки сторінок матиме брошура формату А5, яка містить 3000 символів тексту розміру 2x3? Скільки фарби піде на її роздруківку на принтері з втратами 3 г/лист при нормі витрати фарби 0,15 г/см²?

Описану брошуру друкують два реальних принтера з втратами (4+/-5%) і (4+/-10 %) г/листок. У якому випадку брошура буде більш витратною?

Варіант 6

Базовий клас і його методи: одиничний світлодіодний індикатор. Характеризується допустимою величиною струму I_{max} і коефіцієнтом світності $coef$, залежним від матеріалу, може визначати яскравість світіння (в умовних одиницях) за величиною струму, що протікає через нього, як добуток $coef$ на корінь з I_{max} , а також зворотнє - за заданою яскравістю світіння визначати необхідний для цього струм і визначати "можна чи ні" отримати бажану яскравість, не перевищуючи величину допустимого струму.

Підклас 1 і його методи: багатосегментний індикатор, складений з подібних N одиничних. Яскравість і струм індикатора в цілому береться, як сума його сегментів.

Підклас 2: на підставі відомого ідеального індикатора будується реальний індикатор з відхиленням величини $coef$ від потрібної на величину в діапазоні (+ / -) %.

Практичні додатки:

Одиничний індикатор з $\text{coef}=0,675$ і $I_{\text{max}}=15$ мА світить з яскравістю 1,354 у.о. Який струм при цьому протікає через нього і чи вийде струм за допустиму межу при спробі збільшити яскравість в 2 і 3 рази ?

З одиничних індикаторів з $\text{coef}=0,675$ і $I_{\text{max}}=15$ мА складено табло - матриця 100x20. Який струм буде споживати матриця в режимах роботи 10, 30, 50 і 75% від допустимої яскравості?

Наскільки буде відрізнятися яскравість світіння п'яти реальних світлодіодів з номіналом $\text{coef}=1,250 \pm 30\%$ і $I_{\text{max}}=30$ мА?

Варіант 7

Базовий клас і його методи: кільце на площині. Обчислення площі по зовнішньому і внутрішньому кордону, а також площі безпосередньо самої фігури.

Підклас 1 і його методи: об'ємне кільце - шайба; обчислення об'єму, обчислення маси.

Підклас 2: об'ємне кільце з випадково заданим розміром.

Практичні додатки:

З листа металу необхідно вирубати шайби розміром 5x4 мм. Скільки таких шайб вийде з 1 м² круглої заготовки, якщо коефіцієнт нещільності розміщення прийняти рівним 0,85? Скільки шайб вийде з 2,5 і 5 м²?

Яка буде маса отриманих партій шайб і яка маса металу піде у відходи в кожному випадку? Матеріал для кожного випадку відповідно: сталь (7800 кг/м³) латунь (8500 кг/м³) і папір (1000 кг/м³), а товщина відповідно 1,5 мм.

У коробці знаходиться суміш з 300 латунних і 200 сталевих шайб довільних розмірів: зовнішній діаметр - 2...20 см, а внутрішній - 80 % від зовнішнього. Яка сумарна маса вмісту коробки?

Варіант 8

Базовий клас і його методи: моделювання процесу засіву с/г поля розміром (x, y) насінням з нормою висадки $norm$. Обчислення витрати посадкового матеріалу за площею, зворотно - необхідної площі виходячи з маси насіння і перевірка можливості засіву "так/ні" конкретної площі.

Підклас 1 і його методи: облік % сходів насіння та % його псування в процесі висадки через недосконалість техніки.

Підклас 2: моделювання реальної с/г техніки, що відрізняється від ідеальної величиною відхилення параметра "% псування насіння" у вказаному діапазоні (+ / -) %.

Практичні додатки:

Скільки кг насіння необхідно для засіву поля 2,5 x 4,8 км при нормі витрати 0,24 кг/м²?

Якої площі має бути поле для висадки на ньому 3000 кг насіння з 85 % сходів і нормою витрати 0,32 кг/м², якщо поле засаджує сівалка з 5% показником псування насіння ?

На кожне з двох полів розмірами 1,5x7,3 км було висаджено 1500 кг насіння з 70 % сходів відповідно і нормою витрати 0,32 кг/м². У якому випадку було витрачено більше посадкового матеріалу якщо 1-е поле засаджує сівалка з показником псування насіння 5% +/-10 %, а 2-е - сівалка з показником 5% +/- 20 %?

Варіант 9

Базовий клас і його методи: прямокутник на площині; обчислення площі і периметра прямокутника.

Підклас 1 і його методи: паралелепіпед (об'ємний прямокутник); обчислення об'єму та площі бічної поверхні.

Підклас 2: паралелепіпед з випадково заданими розмірами.

Практичні додатки:

Визначте яка повинна бути прямокутна листова заготовка для того, щоб з неї можна було виготовити 37 однакових прямокутних деталей. Коефіцієнт щільності розміщення прийняти рівним 0,82. Аналогічні дії повторіть для 70, 75, 100 і 115 деталей.

Перевірте, чи зможуть розміститися на аркуші розміром 1м*1м 8-м прямокутників випадкового розміру зі сторонами в діапазоні 17...23 см? Скільки таких прямокутників необхідно для максимального заповнення площі аркуша?

Паралелепіпед з відношенням площі основи до товщини, як 1:3 пофарбували зеленою фарбою з боків і жовтою обидві основи. Аналогічно вчинили з прямокутником 3:1. У якому випадку, якої фарби і в скільки разів більше знадобилося?

Варіант 10

Базовий клас і його методи: точка, що рухається прямолінійно з постійною швидкістю і з координатою X_0 в початковий момент часу. Визначення пройденої за час t відстані, обчислення координати нового положення точки через час t , а також визначення часу необхідного для проходження заданої відстані.

Підклас 1 і його методи: точка, що рухається з прискоренням A . Пройдений шлях враховує додаткову складову $a*t^2/2$.

Підклас 2: на основі переданої точки створює таку, яка відрізняється від неї тим, що початкова швидкість змінюється на величину зазначеного відхилення в діапазоні (+/-)%.

Практичні додатки:

Два поїзди почали рух з точок з координатами 0 і 3 км від початку відліку зі швидкостями 75 і 85 км/год в одному напрямку. Простежуючи зміну їх координат з інтервалом в 7 хв визначте координату точки обгону і відповідний їй момент часу.

Рухаючись в парі зі швидкістю 18 км/год два спринтери почали прискорюватися на спуску, причому 1-й з прискоренням 5 км/год², а 2-й - 4 км/год². Визначте розрив між ними через 3 хвилини.

На відстані 275 метрів знаходиться ціль, для знищення якої було випущено два снаряди. Нормована швидкість вильоту снаряду становить 300 км/год (+/-) 5 %. Визначте, який снаряд швидше влучить в ціль?

Контрольні питання

1. Поясніть використання ключового слова "extends".
2. Як конструктори підкласів можуть викликати методи суперкласу?
3. Поясніть суть використання перевизначених методів?
4. Яка різниця між перевантаженим та перевизначеним методом?

ЛАБОРАТОРНА РОБОТА №4

« Створення UML-діаграми для відображення абстрактних класів та їх реалізації».

Мета: Зрозуміти ідеологію та прийоми побудови ієрархії з використанням абстрактного класу, навчитися описувати його конкретні реалізації і працювати з масивами типу абстрактного класу.

Рекомендації до виконання роботи:

Особливості побудови та реалізації абстрактних класів

Особливістю абстрактних класів являється наявність у їх складі хоча б одного абстрактного методу, тобто такого методу, реалізація якого не розписується, а вказується тільки його назва та тип значення, що повертається, якщо таке є. Водночас, в такому класі можуть міститися поля і конкретні методи, які, також як і абстрактні, будуть успадковані класами нащадками.

Абстрактний клас не може мати представників - для цього його необхідно наслідувати в підклас в якому запропонувати конкретну реалізацію абстрактних методів. У зв'язку з цим у складі абстрактного класу немає конструкторів.

Необхідність використання абстрактних класів виникає тоді, коли створюються класи, не пов'язані між собою спадкуванням "від загального до приватного", які стоять ніби на одному рівні ієрархи, що володіють однаковим набором полів і методів, але в той же час по-різному виконують ті або інші дії. У такому випадку ці класи повинні наслідувати один абстрактний суперклас, до складу якого включено весь набір загальних для них компонентів і задекларовані ті методи, які повинні бути перевизначені у кожному підкласі по-різному залежно від характеру описуваної ним моделі.

Побудована за таким принципом ієрархія дає можливість об'єднувати підкласи на рівні абстрактного класу, наприклад створити масив типу абстрактного класу, елементи якого можуть мати тип підкласів. Тепер для кожного елемента масиву можна викликати той самий перевизначений в підкласах метод, оскільки його ім'я і значення, що повертається, однаково в кожній приватній реалізації.

Приклад побудови ієрархії з використанням абстрактного класу

Нехай необхідно визначати масу рідини в об'ємі ємностей різної форми. Очевидно, що ємності різної форми мають різний об'єм, причому вираховується він по-різному, залежно від параметрів, що характеризують його форму, в той час як маса наливої рідини в будь-якому випадку буде визначатися, як добуток щільності рідини на об'єм ємності (у випадку якщо вона заповнена на 100%).

Логічно було б винести метод "обчислення маси" на рівень абстрактного класу, метод "визначення об'єму" задекларувати в ньому як абстрактний і успадковувати від цього класу необхідну кількість підкласів, в

яких реалізувати визначення об'єму, наприклад, для циліндричної та паралелепіпедної форми ємності.

```

abstract class someContainer {
    double getMass (double density) {
        return density * getCapacity ();
    }
    abstract double getCapacity ();
}

class cylynderContainer extends someContainer {
    double diametr;
    double height;
    cylynderContainer (double diametr, double height) {
        this.diametr = diametr;
        this.height = height;
    }
    double getCapacity () {
        return Math.PI * diametr * diametr * height / 4;
    }
}

class paralelContainer extends someContainer {
    double x;
    double y;
    double height;
    cylynderContainer (double diametr, double height) {
        this.x = x;
        this.y = y;
        this.height = height;
    }
    double getCapacity () { return x * y * height / 4;
}
}

```

/* Тепер можна в тестовому класі створити масив ємностей різної форми і порахувати масу наливої в них води, спирту і т.д.

```

*/
someContainer sosl [ ] = new someContainer [3];
sosl [0] = new paralelContainer (5, 5, 20);
sosl [1] = new cylynderContainer (4, 10);
sosl [2] = new cylynderContainer (4.5, 10);
double sum = sosl [0]. getMass (1.254) +
sosl [1]. getMass (2.354) +      sosl [3]. getMass (0.724);
System.out.println ("Total mass of liquid =" + sum);

```

Приклад побудови ієрархії з використанням абстрактного класу

Припустимо необхідно визначити відносну оціночну характеристику об'єму рідини, що налита в ємність як "менше половини", "приблизно половина", "більше половини" і "під самий верх". У даному випадку маємо 3 діапазони: 0 ... 45%, 45 ... 55%, 55 ... 95% і 95 ... 100%, де за 100% взято повний об'єм ємності, що повертається методом getCapacity ().

У даному випадку можна додати до складу абстрактного класу додаткове поле "реально наливо" і реалізований метод нормування, який би брав величину об'єму наливої рідини і повертав її процентне відношення до повного об'єму ємності, а також абстрактний метод виводу на екран інформації про оцінювання, який буде різним для ємностей тому, що число діапазонів і принцип розбиття по них для кожного може бути свій.

```

abstract class someContainer {
    double realNalito;
    double toNormal (double nalito) {
        return 100 * nalito / getCapacity ();
    }
    abstract void showHowmuch ();
}

```

// Останнє без зміни

```
}
```

У кожному підкласі опишемо свою реалізацію абстрактного методу:

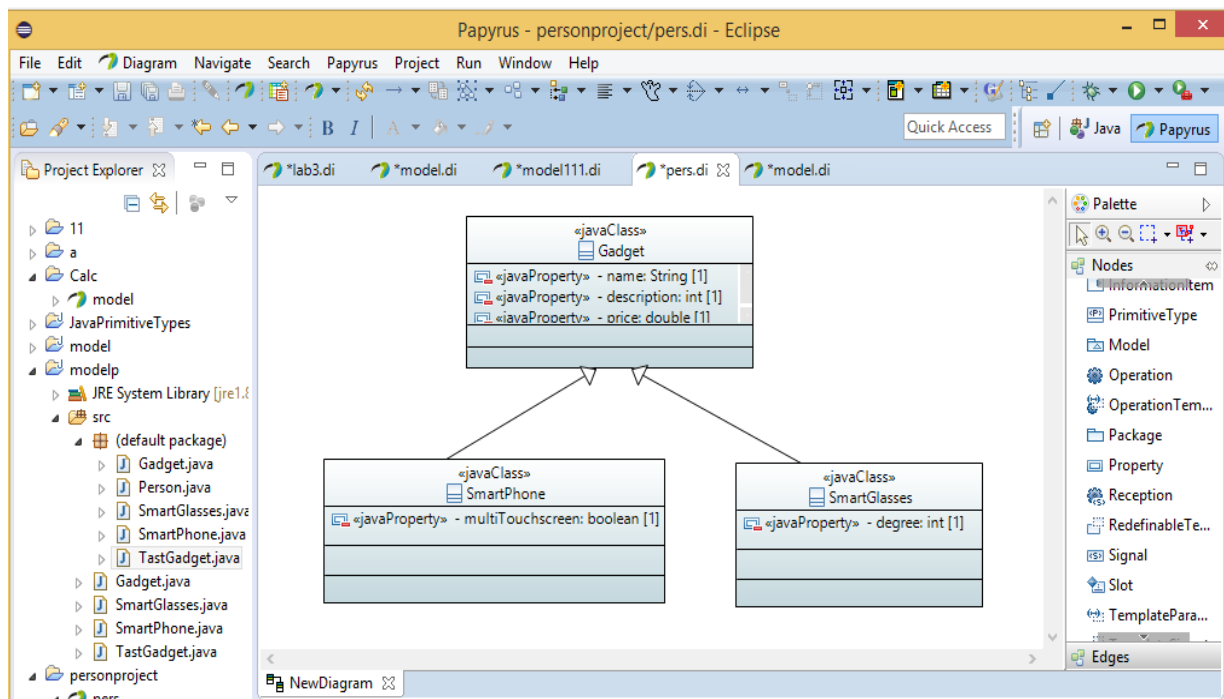
```
void showHowmuch () {
    String s;
    double n = toNormal (realNalito);
    if (n >= 95) s = "під самий верх";
        else if (n >= 55) s = "більше половини";
        else if (n >= 45) s = "приблизно половина";
        else * s = "менше половини";
    System.out.println (s);
}
```

Тепер можна в тестовому класі перевірити роботу методу, перебравши в циклі весь діапазон об'єму:

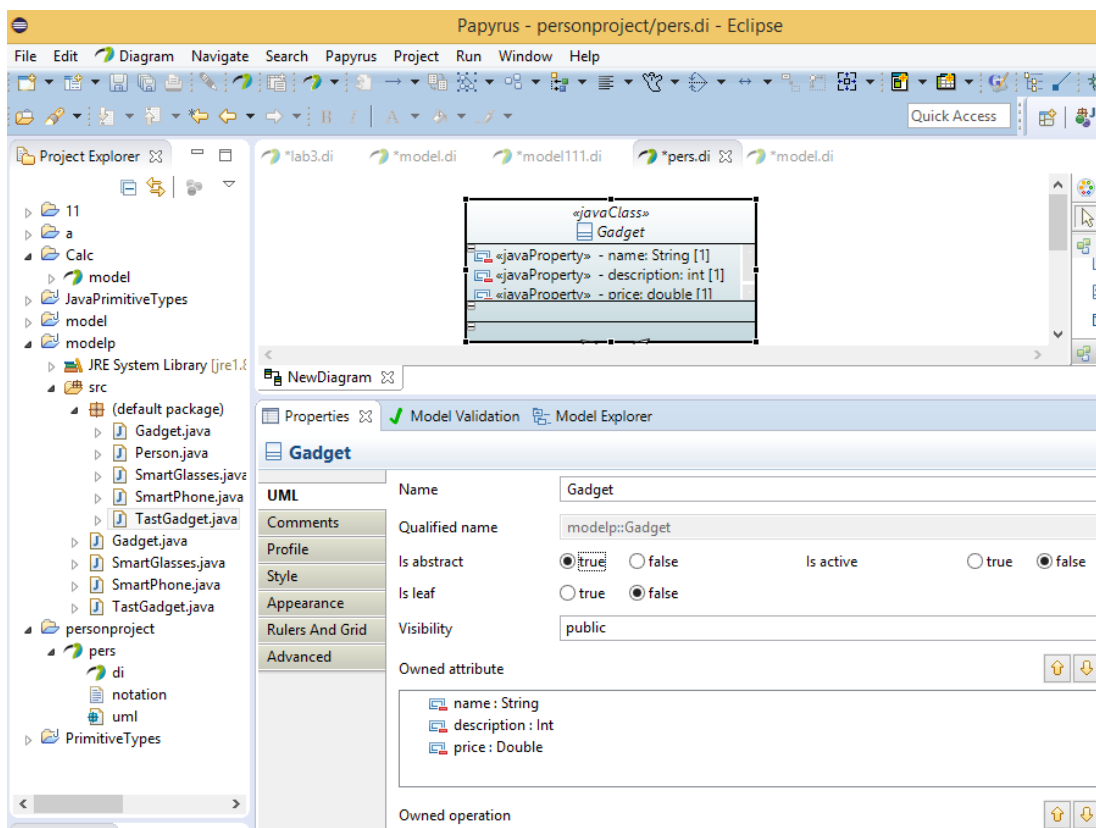
```
sos = new cylinderContainer (5, 20);
double step = sos.getCapacity () / 100;
sos.realNalito = 0;
while (sos.realNalito < sos.getCapacity ()) {
    showHowmuch ();
    sos.realNalito = sos.realNalito + step;
}
```

Рекомендації до виконання роботи (частина 2): моделювання за допомогою плагіну Rapurys.

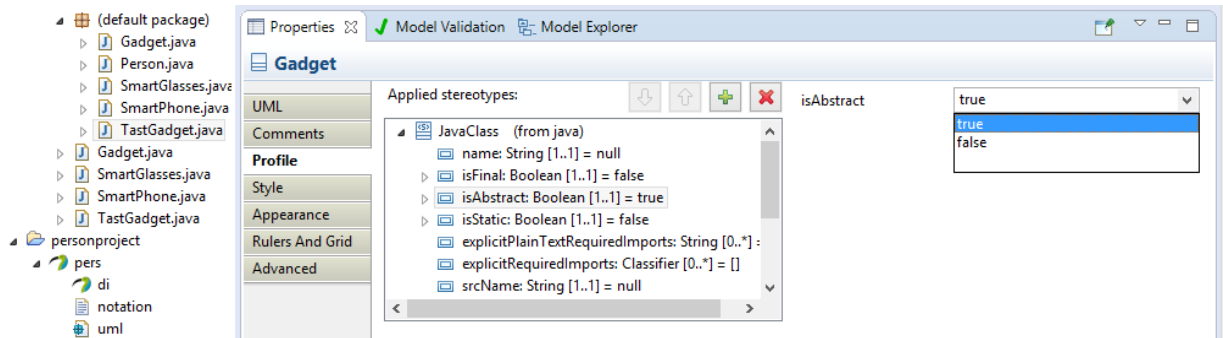
Розглянемо діаграму класів з минулого лабораторної роботи (№3)



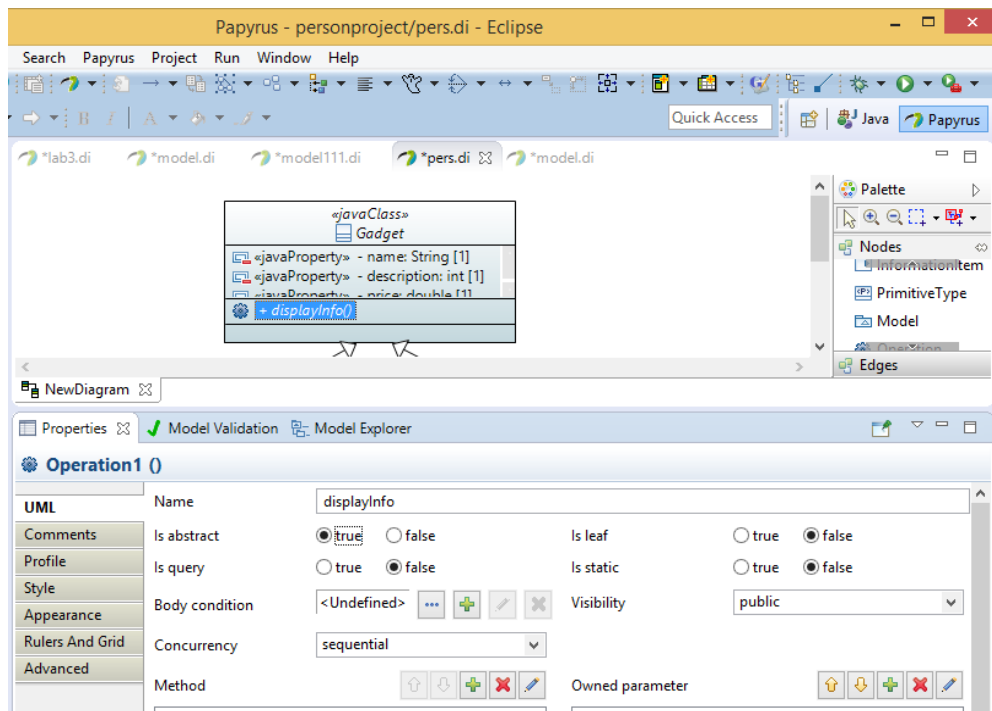
Тепер оголосимо клас Gadget як абстрактний. Для цього в закладці UML активуємо властивість «abstract» = true. В результаті ім'я класу буде відображено курсивом.



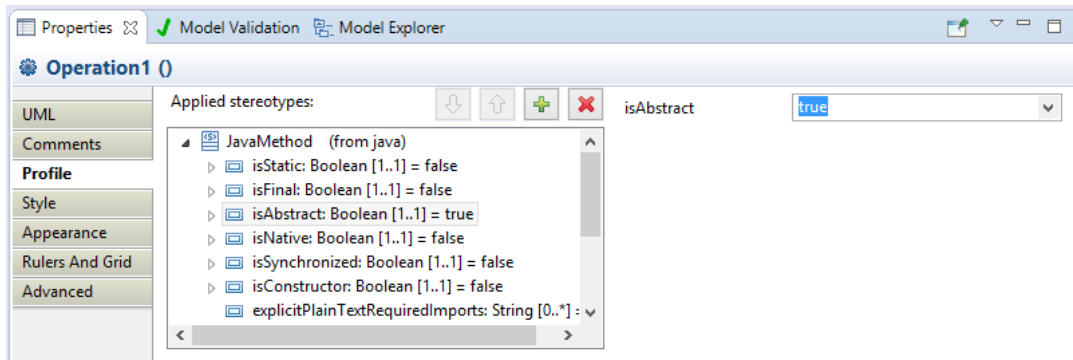
Потім переходимо на вкладку Profile (Properties), де активуємо стереотип Abstract: Boolean [1..1] = true наступним чином:



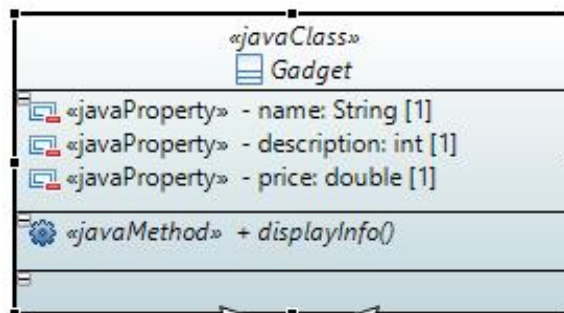
Так як у абстрактного класу повинен бути хоча б один абстрактний метод – об’явимо його як `displayInfo()` у класі `Gadget`. Активуємо для даного методу стереотип `Is abstract = true` на закладці `UML`.



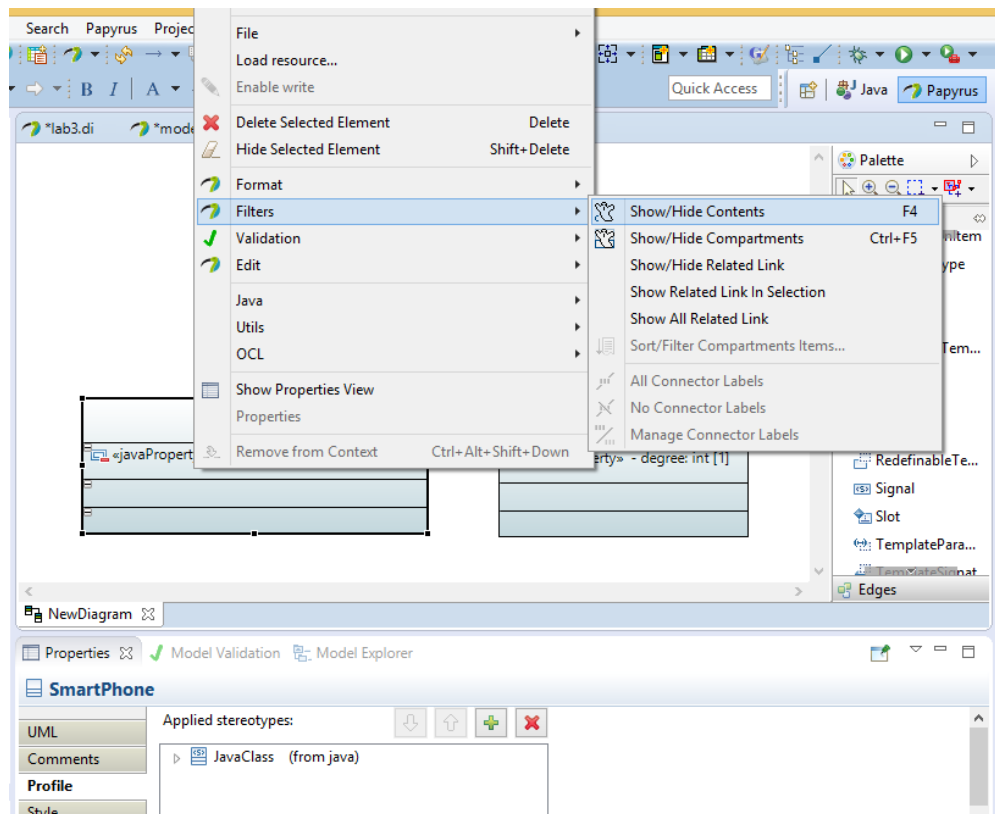
Крім цього на закладці Profile зі списку `JavaMethod (from Java)` потрібно обрати стереотип `isAbstract: Boolean [1..1] = true`



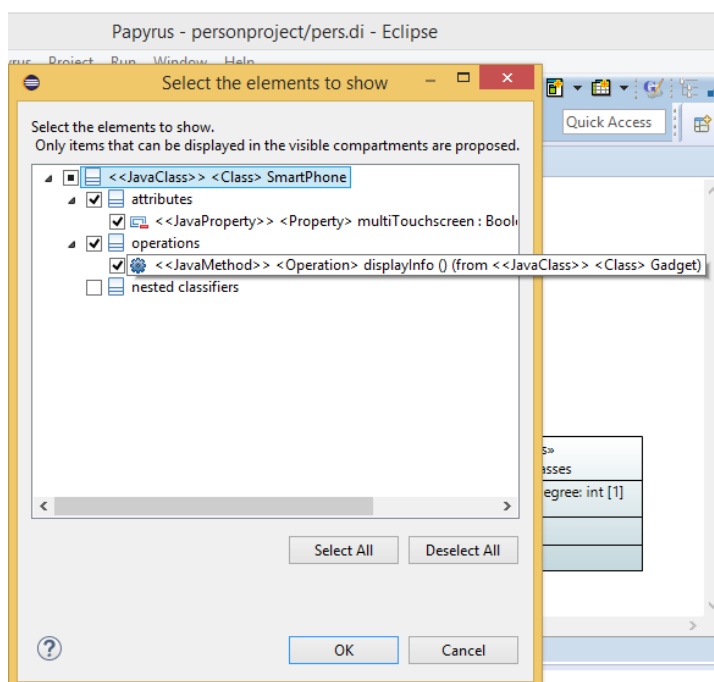
В результаті наш клас на діаграмі буде мати наступний вигляд:



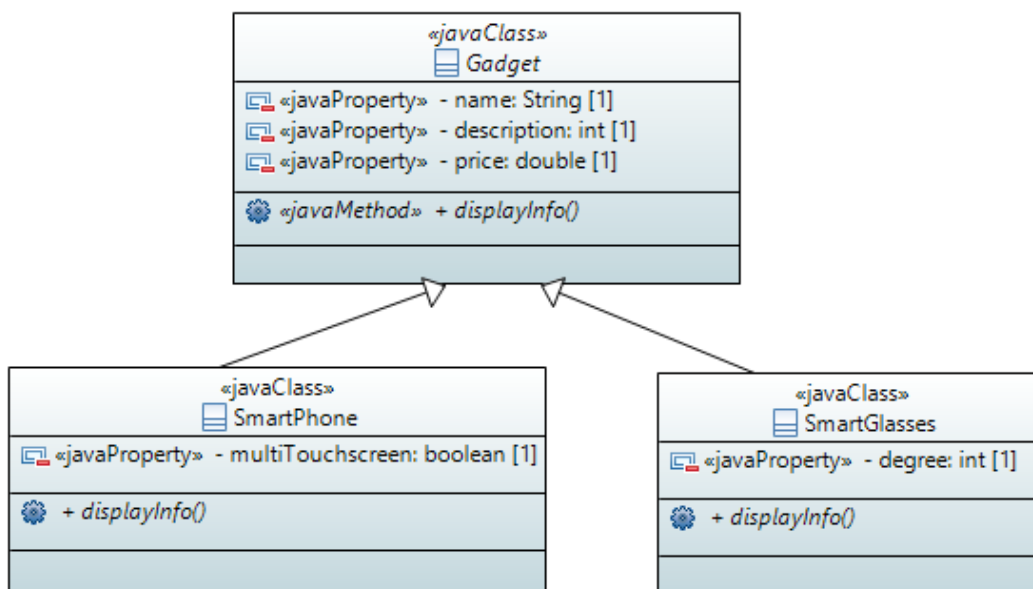
Для кожного з класів-нащадків необхідно встановити фільтр для видимості абстрактного методу класу-батька. Правим кліком по класу-вибираємо filter-> Show / Hide Contents



Відзначаємо галочкою абстрактний метод класу-батька.



В результаті класи-нащадки на діаграмі будуть мати наступного вигляду:



Сгенерований Java – код для нашої діаграми класів prime наступний вигляд:

```

public abstract class Gadget {

    /**
     *
     */
  
```



```
private String name;
/**
 *
 */
private double price;
/**
 *
 */
private int description;
/**
 * Getter of name
 */
public String getName() {
    return name;
}
/**
 * Setter of name
 */
public void setName(String name) {
    this.name = name;
}
/**
 * Getter of price
 */
public double getPrice() {
    return price;
}
/**
 * Setter of price
 */
public void setPrice(double price) {
    this.price = price;
}
/**
 * Getter of description
 */
public int getDescription() {
    return description;
}
/**
 * Setter of description
 */
public void setDescription(int description) {
    this.description = description;
}
/**
 *
 */
public abstract void displayInfo();
}
```

Клас-нащадок SmartPhone:

```
public class SmartPhone extends Gadget {

    /**
     *
     */
    private boolean multiTouchscreen;

    /**
     * Getter of multiTouchscreen
     */
    public boolean getMultiTouchscreen() {
        return multiTouchscreen;
    }

    /**
     * Setter of multiTouchscreen
     */
    public void setMultiTouchscreen(boolean multiTouchscreen) {
        this.multiTouchscreen = multiTouchscreen;
    }

    /**
     *
     */
    public void displayInfo() {
        // TODO Auto-generated method
    }
}
```

Постановка завдання і вихідні дані для варіантів:

Завдання:

Необхідно описати абстрактний клас criterion, що має в своєму складі:

1. оголошення загального для ієрархії набору полів;
2. реалізовані методи:
3. нормування: приймає поточне значення параметра і його максимальне значення, ділить перше на друге і повертає величину в % щодо прийнятого за 100%. Працює з числами типу double;

4. інші необхідні методи виходячи з конкретного варіанта;
5. абстрактні методи:
6. розрахунок оціночного критерію: працює з полями конкретного класу і залежно від них за специфічними законами даного класу виробляє деякий інтегральний показник стану даного об'єкта і повертає його, як величину типу double;
7. порівняння відносно: розглядає величину нормованого показника на предмет приналежності певного діапазону і виводить на екран текстову оціночну характеристику цього діапазону.

Опишіть класи `criterionRealiz1`, `criterionRealiz2` і `criterionRealiz3`, які є реалізаціями абстрактного класу, де кожен з них:

1. додає в свій склад необхідні поля і відповідні конструктори;
2. реалізує абстрактні методи;
3. по необхідності додає додаткові специфічні методи.

Опишіть тестовий клас `testing`, в якому створіть необхідну кількість представників неабстрактного класу і з допомогою них розв'яжіть конкретне практичне застосування.

Зауваження: реалізації абстрактного методу "розрахунок оціночного критерію" має чисто умовний характер - це комбінація чисельно-логічних виразів (полів об'єктів) повертає величину типу `double`, правильність якої з наукової точки зору не розглядається.

Варіант 1

Група студентів з 3 чоловік складала іспити. Студенти отримали відповідно оцінки 0...+5 балів кожен, причому оцінка кожному студенту ставилася з урахуванням наступних показників: % відвідуваності лекцій, %

відвідуваності лабораторних занять, наявність конспекту, % правильних відповідей на екзаменаційний білет, впевнений тон і особиста чарівність.

На підставі даних показників (індивідуальних для кожного студента) за деякою залежністю (різною для кожної групи), відомою екзаменатору (і людині, яка дану задачу) вироблявся якийсь інтегральний показник знання і далі шляхом порівняння з 5-ти бальною шкалою виставлялася оцінка ("відмінно", "добре" і т.д.). Промоделюйте цю ситуацію і визначте найбільш успішного з них.

Варіант 2

На будівельному майданчику три бригади землекопів риють траншеї для заливки фундаменту під новий оптово-роздрібний магазин. Кожен день виконроб контролює пророблений об'єм робіт і при виконанні якої-небудь бригадою чергового етапу роботи доповідає керівнику проекту у вигляді "Бригада номер N закінчила ("підготовчий", "початковий", "основний", "завершальний", "фінальний") етап роботи.

Продуктивність $\text{м}^3/\text{день}$ кожної бригади залежить від кількості осіб в ній, коефіцієнта справності ручного інструменту, % застосування автоматизованих засобів і відсутність дощів і розраховується по різному для кожної бригади. Об'єм траншей у кожній бригади різний.

Промоделюйте ситуацію від одночасного початку роботи бригад до того моменту, як сама остання закінчить свою роботу, формуючи при цьому (при необхідності) добові рапорти виконроба.

Варіант 3

Рухомий бажанням правильно вибрати майбутню спеціальність, абітурієнт пройшов 3 тести на наявність у нього схильності до програмування, математики та економіки, відповівши на 10-ть питань кожний і отримуючи за кожну відповідь "так" - два, за "ні" - нуль і за "не знаю" - один бал.

Після проходження тесту бали необхідно обробити по деякій залежності (для кожного тесту різна), що включає в себе номер питання, їх загальну кількість і фактор широти відповідей, отримавши таким чином підсумкових критерій оцінювання, після чого проводиться розшифровка тесту залежно від того, в якій з 6-ти діапазонів потрапив даний критерій.

Промоделюйте відповіді, задавши їх випадково або безпосередньо, і визначте якою мірою абітурієнт може вважати себе потенційним програмістом, математиком або економістом, вивівши на екран відповідну отриманому балу розшифровку тесту для кожного.

Варіант 4

Фірма, що займається вантажоперевезеннями, проводить розрахунок необхідного об'єму бензину для поїздки в залежності від наступних показників: маса перевезеного вантажу, наявність причепа, коефіцієнт захищеності вантажу від вібрацій, характер небезпеки вантажу, коефіцієнт рельєфу місцевості і дальність шляху.

На підставі цих параметрів по емпірично встановленій нормувальниками залежності (для кожного маршруту своя) визначається інтегральний коефіцієнт бензинозатратності поїздки. Три машини вирушають в дорогу, кожна на різну відстань в свій регіон і зі своїм вантажем. Максимальна відстань, охоплена діяльністю фірми, 500 км.

Визначте для кожної машини бензинозатратність її поїздки за 6-ти бальною шкалою "майже біля будинку», «не далі сусіднього села", "як до шефа на дачу", "до вечора будемо на місці", "відрядження не менше, ніж на 2 дні", "шліть гроші на зворотну дорогу" (характеристики дано умовно) і визначте, який машині належить найбільш важкий рейс.

Варіант 5

Виноробний завод "Бадьорий дух" відправляє на щорічну виставку дегустацію свою продукцію в надії на розширення ринку збуту. Продукція представлена трьома сортами вин.

Конкурсна комісія приймає рішення про присудження призів на підставі комплексного показника якості продукту, який вираховується по деякій залежності, встановленій для різних сортів вин виноробами минулих століть, на підставі наступних показників: % вмісту цукру, % вмісту спирту, наявність сторонніх домішок, термін витримки, оцінка досвідчених дегустаторів, оцінка любителів дегустацій.

Промоделюйте продукцію заводу і визначте, які сорти потрапили в категорії "кращий важко уявити", "гідний продукт гідного конкурента", "для широких верств населення", "для тих хто не відчуває смаку", "а це взагалі що за напій?" (характеристики дано умовно), прийнявши за максимум показник минулого року.

Варіант 6

Приватна фірма займається розфасовкою чаю, отримуючи сорти, які умовно розділяються співробітниками фірми за категоріями "чисто вищий сорт", "вищий сорт", "типу вищий сорт" і "сорт вищий" на підставі виведеного ними комплексного показника якості (залежно від дня тижня) на підставі таких чинників: % вмісту безпосередньо чаю, % вмісту сировини подібної до чаю, % сировини, яка до чаю відношення не має, але смак не погіршує, наявність факторів розкрадання компонентів сировини та наявність гризунів у виробничих приміщеннях. На протязі трьох днів роботи був розфасований чай різних сортів.

Визначте який сорт був відвантажений у кожен з днів і якого сорту вийшло більше.

Варіант 7

В деякій країні за результатами перепису населення було вирішено визначити якийсь сукупний показник благополуччя з демографічної ситуації в трьох областях, причому були прийняті наступні показники благополучності: % народжуваності, % смертності, рівень безробіття, рівень соціальної захищеності, наявність небезпечних і шкідливих виробництв, частота природних катаклізмів. Характер обробки даних індивідуальний для кожної області.

Промоделюйте дану ситуацію і визначте найбільш і найменш благополучну з них. Оцініть найбільш вдалу область за шкалою "гірше нікуди", "жити можна", "могло б бути і краще", "цілком нормально", "дуже навіть непогано", "краще вже нікуди" (градації дано умовно) з трьома сусідніми, прийнявши в кожному випадку її рівень благополуччя за максимально можливий.

Варіант 8

У надії заробити собі на гідну старість, у верхів'ях золотоносної річки ведуть роботу три артілі старателів, причому кожна з них працює зі своєю продуктивністю, що вираховується (методом, індивідуальним для кожної артілі) на підставі таких чинників: % золота в піску, кількість працівників, регулярність постачання артілі продуктами та витратними матеріалами, психологічний клімат у колективі, наявність диких звірів у зоні розробки.

Який час знадобитися кожної артілі для досягнення мети (10 кг кожному) і хто в даний момент працює краще за всіх, а хто "приблизно так само", "трохи гірше, але теж непогано", "повернеться додому ще цього року", "залишиться ще на рік", "залишиться винний і немало".

Варіант 9

Клуб любителів собак «Фаворит» проводить щорічні монопородні виставки собак. Змагання проводяться за трьома породам. Комісія приймає рішення про присудження призів і звань на підставі комплексного показника відповідності, який базується на підставі наступних показників: вік собаки, висота в холці, вага, форма голови, якість шерсті, оцінка по дресурі.

Промоделуйте результати змагань і визначте які породи собак потрапили в категорії «краща мисливська собака», «краща службова собака», «сама компактна собака для малогабаритних квартир», «краща собака-нянька для сім'ї з дитиною».

Варіант 10

Кондитерський цех «Буратіно» почав виробництво низькокалорійних тортів, що розділяються за категоріями: «для вживання після 18.00», «для страждаючих діабетом», «чиста хімія», «прощавай фігура» на підставі свого комплексного показника якості, який визначається наступними показниками: % жирів, % вуглеводів, % моно-і дисахаридів, наявність ароматизаторів і барвників, наявність шкідливих харчових добавок.

Торти випікали три дні. Визначте, якої категорії тортів спекли найбільше?

Контрольні питання

1. В чому полягає особливість абстрактних класів?
2. Чому у складі абстрактних класів не має конструкторів?
3. Коли виникає необхідність у використанні абстрактних класів?

